

AD-A070 850

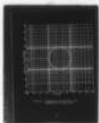
CIVIL ENGINEERING LAB (NAVY) PORT HUENEME CA
USER'S MANUAL FOR FVSOLVR AND FVPLT: FIELD SOLVING AND PLOTTIN--ETC(U)
MAR 79 B R MILNER
CEL-TN-1549

F/G 9/2

UNCLASSIFIED

NL

1 OF 1
AD
A070850



END
DATE
FILMED
8 79
DOC

LEVEL *II*

12

Technical



Note

TN no. N-1549

title: USER'S MANUAL FOR FVSOLVR AND FVPLT: FIELD SOLVING AND PLOTTING COMPUTER PROGRAMS

author: Brian R. Milner

date: March 1979

sponsor: Naval Facilities Engineering Command

program nos: YF53.534.091.01.201



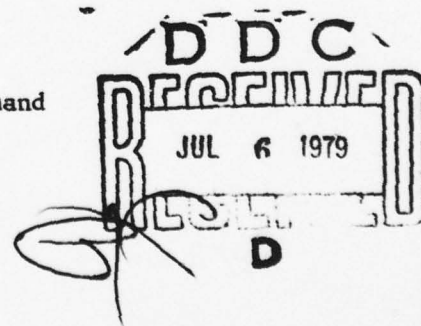
CIVIL ENGINEERING LABORATORY

NAVAL CONSTRUCTION BATTALION CENTER
Port Hueneme, California 93043

Approved for public release; distribution unlimited.

AD A070850

DDC FILE COPY



9 Technical note Sep 75-Dec 78

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TN-1549	2. GOVT ACCESSION NO. DN344001	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) USER'S MANUAL FOR FVSOLVR AND FVPLOT: FIELD SOLVING AND PLOTTING COMPUTER PROGRAMS		5. TYPE OF REPORT & PERIOD COVERED Not final; Sep 75 - Dec 78
7. AUTHOR(s) Brian R. Milner		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS CIVIL ENGINEERING LABORATORY Naval Construction Battalion Center Port Hueneme, California 93043		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Facilities Engineering Command Alexandria, Virginia 22332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62760N; YF53.534.091.01.201
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) CEL-TN-1549		12. REPORT DATE March 1979
		13. NUMBER OF PAGES 74
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) YF53534		
18. SUPPLEMENTARY NOTES YF53534091		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Electric field, equipotential, potential distribution, computer program, computer plotting, dielectric, insulator.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) FVSOLVR and FVPLOT are, respectively, computer programs which solve and plot field distributions for problems that involve the solution of Laplace's equation. The programs were developed primarily for solving electric field and equipotential distributions, but may be adapted to solve problems such as temperature distributions or problems which involve the solution of Poisson's equation. They can also be used with either axisymmetric or two- dimensional geometries. This user's manual provides the detailed (continued)		

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391 111

503

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Continued

information required to implement problem solutions with these programs.

Library Card

Civil Engineering Laboratory
USER'S MANUAL FOR FVSOLVR AND FVPLOT: FIELD
SOLVING AND PLOTTING COMPUTER PROGRAMS
by Brian R. Milner
N-1549 74 pp illus March 1979 Unclassified

1. Computer plotting 2. Equipotential distribution I. VF53.534.091.01.201

FVSOLVR and FVPLOT are, respectively, computer programs which solve and plot field distributions for problems that involve the solution of Laplace's equation. The programs were developed primarily for solving electric field and equipotential distributions, but may be adapted to solve problems such as temperature distributions or problems which involve the solution of Poisson's equation. They can also be used with either axisymmetric or two-dimensional geometries. This user's manual provides the detailed information required to implement problem solutions with these programs.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

	Page
INTRODUCTION	1
DESCRIPTION OF PROGRAM FVSOLVR	1
PROBLEM SOLUTION USING FVSOLVR	2
Statement of Problem	2
Development of Data	3
Checking Input Data	11
Output From FVSOLVR	13
USE OF THE PLOT PROGRAM FVPLLOT	13
APPENDICES	
A - Example Problem	33
B - Program Listings	43

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

DDC
RECEIVED
 JUL 6 1979
 D

INTRODUCTION

FVSOLVR is a two-dimensional finite difference field solving algorithm for the calculation of electric field and potential distribution. The nodal solutions from FVSOLVR can be used to plot electric field lines and/or equipotential lines using the CALCOMP plotter and the companion program FVPLOT.

These programs were developed as part of a Navy research effort concerning insulators employed in large antenna arrays. These antenna insulators are subjected to severe electrical field conditions which occasionally result in arcing, corona, and heating or burning of the insulators. In order to determine causes of insulator failures and evaluate alternative designs it is helpful to know the electric field distribution associated with each insulator.

The general capabilities of the programs, the theoretical foundation, and a discussion of possible applications can be found in the Civil Engineering Laboratory Technical Note N-1502, "Computer Technique for Calculation of Potential in Multidielectric Media," by Kwang-Ta Huang, Brian R. Milner, and Andrew W. McClaine. The intention of this manual is to provide a user of FVSOLVR and FVPLOT with the detailed information required in order to implement problem solutions with these programs.

The programs described herein are written for use with a Fortran IV compiler. The program versions listed in Appendix B have been used successfully on the CDC 7600 computer at the Lawrence Berkeley Laboratory, Berkeley, California. Users are cautioned that minor modifications may be necessary if these programs are used with other systems.

DESCRIPTION OF PROGRAM FVSOLVR

FVSOLVR solves problems which are geometric in nature. The geometry of the solution region can be described using either a cartesian coordinate system or a cylindrical coordinate system. The solution region in the cartesian system is a rectangle with a side parallel to the x-axis. In Figure 1, the x-axis can be either of the two axes shown and the solution region may generally include parts of any of the quadrants. (Some versions of FVSOLVR which are designed to take advantage of the symmetry of certain problems require that the rectangular solution region be entirely within the first quadrant.) The rectangle describes a plane in space, perpendicular to the z-axis, with z coordinate z_0 . The condition required for this problem to be two-dimensional is that, as z varies, the field solution does not vary. That is, any similar rectangle translated in the plus or minus z direction will have the same

solution. See Figure 2. An example of such a problem might be a capacitor plate within a housing, as shown in the cross-sectional view of Figure 3. The z-axis is perpendicular to the plane of the paper. The geometry in any plane parallel to the paper must have the same configuration. Idealizations usually must be made in cases such as this since the capacitor plate and housing do not have infinitely long z dimensions. But if d is the largest gap dimension and the actual z dimension of the capacitor is greater than 20d, then experience dictates that the two-dimensional solution is usually a good approximation, until one gets closer than 20d to a boundary in the z direction.

The cylindrical coordinate system, illustrated in Figure 4, can also be used. The solution region is either a cylinder of radius $r = r_1$, or a cylindrical section from $r = r_0$ to $r = r_1$. In either case, the solution region has a height, h, which is measured in the z direction. The z-axis must be an axis of symmetry; that is, the geometry and solution are two-dimensional and independent of θ . The geometry can vary with r and z and the function to be solved is $\phi = \phi(r, z)$, with $\partial\phi/\partial\theta = 0$.

The program incorporates the useful characteristics of variable grid spacing and variable dielectric properties, thus allowing multi-dielectric medium problems to be solved with economy of computer time. The program also has the capability of solving anisotropic media problems. Greater detail about the inner workings of the computer program and relevant electromagnetic theory can be found in CEL Technical Note N-1502.

Input data for the programs are developed and entered in the same way for both coordinate systems. When using the cylindrical system, the z-axis must be an axis of symmetry. Figure 5 illustrates an example of the solution region in the cylindrical case. The known circular boundary in Figure 5 will be explained later. The r-axis may or may not have a plane of symmetry passing through it perpendicular to the z-axis. In the cartesian system, the y-z plane and/or the x-z plane may or may not be planes of symmetry. Input data include grid coordinates, dielectric values (which can vary throughout the grid system), dielectric material boundaries, and known potentials or electric field conditions. Development of program input data will be shown by example, using the cylindrical coordinate system and solving an electric potential distribution problem.

PROBLEM SOLUTION USING FVSOLVR

Statement of Problem

A potential field solution for the region surrounding two parallel corona rings is required. In this problem, a potential is applied across the air gap separating the corona rings, which are metal toroids. See Figure 6. Each toroid has an outside diameter of 20 inches and an inside diameter of 10 inches. Each toroid is suspended by a pipe attached to a thin sheet of metal welded to the center of the toroid. In this case, the z-axis coincides with the center line of the pipes and is

perpendicular to the planes of the toroids. As an example of an application, in Appendix A of this manual, the FVSOLVR program is used to solve this particular problem. The top ring and pipe consists of conducting material at a potential of +100 volts, and the bottom ring and pipe likewise, except at a potential of -100 volts. Not only does this problem have rotational symmetry about the z-axis, but it also has symmetry across a plane perpendicular to the z-axis. The plane defined by $(z=0, r)$ is a mirror symmetry plane. Thus, solving the upper right hand quadrant will give solutions for the other three quadrants as well. This solution region is shown in Figure 7. Significant computational advantage is gained when solving just one quadrant. Only one-fourth as many equations need be solved; therefore, computer costs are greatly reduced. Problems such as this, which have mirror symmetry from top to bottom, will have an equipotential line about which this mirror reflection will occur in the two-dimensional solution. The potential of this symmetry line is always half the sum of the upper and lower potentials. Since the upper potential is +100 volts and the lower potential is -100 volts, the voltage at the center ($z=0$) line is 0 volts.

The voltages in any problem can always be normalized so that this central line has a voltage value of zero volts simply by adding or subtracting a constant value from all known voltages. In general, the relative voltage levels can be shifted by any constant value. The FVSOLVR program requires that all voltages be non-negative. If it is desired to solve a problem involving negative voltages, all voltages must be shifted to eliminate the negative values. After the solution is obtained, the values can be shifted back to the original levels to obtain the true solution. The shape of the flux and equipotential lines will not be affected by this kind of manipulation.

Development of Data

The generation of the input data is performed manually, and usually requires the preparation of a good drawing of the solution region. Note that use of graph paper will facilitate the preparation of the drawing. Grid lines are constructed on the drawing, keeping in mind that any type of boundary must consist of straight line segments beginning and ending on nodes of grid lines. The circle which represents the corona ring in Figure 7 will in fact be a series of straight line segments approximating the circle.

At this time, we will examine boundary conditions, which are known at $r=0$ (viz. $\partial V/\partial r=0$) and at $z=0$ (viz. $V=0$). If the grid region does not include the axis of symmetry but begins at some radial value greater than zero, the boundary conditions for that edge of the grid region must be given explicitly. (A means for inputting this data is described in the section which discusses the input variables NPOTS, POT, NLINES, etc.) But as r and z go to infinity, boundary conditions are not known. A means for treating such problems will be presented here. The theoretical basis for this approach can be found in CEL Technical Note N-1502.

The boundary conditions at large values of r and z can be approximated to sufficient accuracy by considering that at large distances the corona ring gap appears like a point gap. The point gap arrangement is a simple geometry which has a well-known field distribution. At large distances the electric flux lines around a point gap form near-perfect circles. Such an electric flux line can be used as the necessary boundary condition at large distances. Thus, at some distance, which experience dictates to be 10 to 20 times the distance of the gap between the rings, a boundary, assumed to be a circle, is introduced. This circular boundary is shown in Figure 8. Inside the circle, the medium is air with the relative dielectric value taken as unity. In the region outside the circle, the dielectric value is taken to be zero. Mathematically, this means that the region outside the circular boundary does not affect the region within the circular boundary. At least two grid lines are needed outside the circular boundary, one in the r direction and the other in the z direction. These two grid lines coincide with the edges of the solution region shown by dashed lines in Figure 8. The circular boundary and zero dielectric obviate the need for explicit boundary conditions at the edges of the solution region. (Again, it might be noted that this circular boundary is a series of straight line approximations.)

Variable grid spacing is used in the program. This is extremely valuable in that proper use of such a scheme can reduce the number of equations which must be solved. Experience is needed to predict where fine grid spacing is needed and where coarse grid spacing will suffice. A "rule of thumb" is that regions in the neighborhood of the gap, where small radii occur, tend to have the highest voltage gradients and therefore need more grid points for good approximations. Regions far from the gap tend to need fewer grid lines for good approximations. Experience with the program and actual experimental results will aid in arriving at optimal grid designs.

Boundary lines between different dielectric regions are constructed within the grid system. These are composed of line segments constructed either horizontally, vertically, or diagonally between adjacent grid points. Regions of known potential can be assigned any convenient dielectric value. This freedom is allowed because, for regions of known potential, the program does not solve equations involving dielectric values. However, for completeness of the data arrays, all regions must be assigned a dielectric value. It is most convenient to assign to a region of known potential a dielectric constant equal to that of an adjacent region. In such a case, there does not exist a dielectric boundary between the two specified regions since the dielectric constants are taken to be the same.

If a problem includes a conducting medium which is of unknown potential, the region occupied by such a medium must be assigned an appropriate dielectric constant value. Satisfactory results can usually be obtained by assigning to the conducting region a dielectric constant which is very much larger than (and nearly infinite with respect to) the relative dielectric constants of the surrounding insulating media.

Since computers cannot represent infinitely large numbers, there is an upper bound on the value which may be used to represent the relative dielectric constant of the region. If the value chosen is too large, it could result in over or underflow problems with the computer during execution of the program. If the value chosen is not large enough, it will not appear "infinite" with respect to the other relative dielectric constants in the solution region of the problem. The symptom of this malady will show up in the potential distribution solution, where the whole of the conducting medium will not be at an equipotential. The optimum dielectric constant value for a conducting medium of unknown potential will be machine dependent and the user should determine a suitable value by trial and error method. Satisfactory results have been achieved with a CDC 7600 computer using a relative dielectric value of the order of 10^8 .

The data deck format will now be explained. Throughout this manual, in the discussions pertaining to computer program variables such as HX(I), HY(J), NX, NY, NEPR, NEPZ, BDR(I,J), BDZ (I,J), etc., any reference to x holds true for r and vice-versa. The same relationship exists between y and z. The information for the data deck can be obtained almost totally from the drawing of the problem.

The first data card has the following information:

NX, NY, NEPR, NEPZ, IFSKIP, IFBDY
Format 6I4

NX is the number of axial grid lines.

NY is the number of radial grid lines.

NEPR is the maximum number of dielectric boundaries that might be encountered in any given row. A row is defined here as a slot between two adjacent radial grid lines. See Figure 9. Note that there must always be a dielectric boundary which coincides with the axial grid line of maximum radial value. For the grid in Figure 9, the proper value of NEPR is four (4), since there are four dielectric boundaries in the Ith row and no other row has more than four.

NEPZ is the maximum number of dielectric boundaries that might be encountered in any given column. A column is considered to be a slot between two adjacent axial grid lines. See Figure 10. There must always be a dielectric boundary which coincides with the radial grid line of maximum axial value. For the grid in Figure 10, the proper value of NEPZ is ten (10). This value is determined by the Jth column, which has more dielectric boundaries than any other column.

IFSKIP is an indicator used to test input data (to be explained later). Briefly, if IFSKIP \neq 0, the program merely reads the input data and sets up the boundary conditions so that the user may verify that the problem has been programmed correctly. If IFSKIP=0, the program executes the solving algorithm.

IFBDY is an indicator which identifies the type of boundary conditions which are to be used for the top (radial) and right-hand (axial) grid boundaries shown as dashed lines in Figure 8. IFBDY=0 indicates that the boundary conditions are to be treated implicitly, as for the circular boundary in the example of Figure 8. This means that it is not necessary to provide the potentials along the top and right-hand boundaries explicitly as input data (i.e., these boundary potentials can be ignored). IFBDY \neq 0 indicates that potentials along the top and right-hand boundaries are given as input data. This is usually done if the solution region happens to be a fine mesh in a section of a previously solved large, rough mesh grid, so that the boundary potentials can be interpolated from the large grid solution. (Further details are given later.)

The next set of information read is HX(I) for I = 1 through NX. HX(I) are the r coordinates of the axial grid lines. The HX(I) must be ordered in increasing magnitude and must all be greater than or equal to zero. HX(I) is always the distance from the axis (line) of symmetry to the Ith axial grid line. The format for each punched card having HX(I) data is 10F8.0. When a decimal point is used, a maximum of 3 digits to the right of the decimal point is taken; digits beyond the third decimal place are rounded off. Units of measure should be scaled to conform with this format.

The next set of information read is HY(J) for J = 1 through NY. The HY(J) are the z coordinates of the radial grid lines. HY(J) must be ordered in increasing magnitude and can be greater than, less than, or equal to zero. The format for HY(J) data cards is also 10F8.0. Again, only the first three digits to the right of the decimal point are taken and additional digits to the right are rounded off.

HX(I) always represents distances measured in the radial direction for cylindrical problems. Axisymmetric problems must always be oriented in this manner. For problems represented in the cartesian coordinate system where no grid boundary coincides with a symmetry plane, the problem can be oriented in any manner with respect to HX(I) and HY(J). However, the program execution will be faster and less costly if NX is less than NY.

The next information that is read is BDR(I,J) for J = 1 through NEPR, for each I from I = 1 through NY-1. This information indicates where dielectric boundaries exist in the radial direction. The program "searches" the rows (radial slots) between adjacent radial grid lines for dielectric boundaries. The first index (I) is the row index, which is the integer value of a given radial slot, starting with 1 at the bottom of the region. The second index (J) is the dielectric index which indicates the Jth boundary encountered in the Ith row when moving outward from the axis of symmetry.

The assignment of the values of BDR(I,J) is explained with respect to the example shown in Figure 11, where NEPR=3 and NY-1=4. If a boundary exists axially (parallel to the axis of symmetry), BDR(I,J) is given a value of two times the index if the axial grid line on which the boundary occurs. If the boundary line is diagonal, it is assigned a value equal

to the sum of the indices of the two axial grid lines between which the diagonal boundary occurs. This will always be an odd integer, whereas all axial boundaries will always have even BDR values. If a given row has fewer than NEPR boundaries, dummy boundaries must be placed at the last axial grid line to satisfy the computer program. Therefore, in this example, NEPR=3 requires that a dummy boundary be placed in row 3 at grid line 6 to complete the BDR(I,J) array. If, for example, NEPR were to equal four and NY-1 were to equal fifty, four dielectric boundaries would have to exist for each one of the fifty rows. It is quite possible that, in many rows, several boundaries will be "stacked" at the last grid line in order to satisfy the program.

Another example of the assignment of the BDR(I,J) values is illustrated in Figure 12. For this case, NEPR=4. The format required for the input of the BDR(I,J) values is 20I4. (Thus, in the present example, all of the BDR(I,J) data would go on one card.) In general, NY-1 groups of NEPR integers are used to describe the boundaries that might be encountered in the radial direction.

The next information read is a series of numbers, KO, KN, KK, and EPSR(K) for K = 1 through KK. These numbers are used to assign dielectric values to the regions identified by the radial boundary search just described. The information is governed by the order of appearance of the dielectrics. For example, in Figure 12, for rows one, two, three and four, respectively, the dielectric values, in the order of appearance are:

Row 1:	1.0	10.0	1.0	
Row 2:	20.0	1.0	10.0	1.0
Row 3:	20.0	1.0	10.0	1.0
Row 4:	20.0	1.0		

A given order of appearance is referred to as the dielectric sequence for the particular slot. KO is the index of the first row in which a new sequence of dielectrics is encountered. KN is the index of the last row having that same sequence, with the provision that if KN-KO is greater than 1, all rows between the KOth and KNth rows also have the same dielectric sequence. KK is the number of dielectrics in the sequence.

The first card to be read for this series of numbers contains KO, KN, KK for the first dielectric sequence. The format is 3I4. This is followed (on the next data card) by the values EPSR(K), for K = 1 through KK, of the dielectric constants in their order of appearance. The format for this data is 16F5.0. This process is repeated for successive dielectric sequences appearing in the solution region (until KN=NY-1). For the example of Figure 12, this part of the data is as follows:

KO,KN,KK	1	1	3
EPSR(K)	1.0	10.0	1.0

KO,KN,KK	2	3	4	
EPSR(K)	20.0	1.0	10.0	1.0
KO,KN,KK	4	4	2	
EPSR(K)	20.0	1.0		

Since the sequence of dielectrics in this example is identical for Rows 2 and 3, the values for these two rows can be read in together. This "shorthand" is quite useful for most problems, where many successive rows will contain the same sequence of dielectrics.

A similar procedure is used for input of boundaries in the axial direction search. The BDZ(I,J) for J = 1 through NEPZ, for each I from I = 1 through NX-1 are read. The rules which govern the BDZ data are analogous to the rules for the BDR data. The program "searches" the columns (axial slots) between adjacent axial grid lines for dielectric boundaries. BDZ(I,J) represents the Jth boundary in the Ith column. There are NX-1 groups of NEPZ boundaries read in format 20I4.

Another set of KO's, KN's, and KK's are then put into the data set with their accompanying EPSZ(K) values. The EPSZ is the dielectric constant value in the radial direction. This value need not be the same as EPSR at the same point, which represents the dielectric constant in the axial direction. This allows anisotropic medium problems to be solved.

Boundaries which are radial do not appear in the BDR(I,J) array. Boundaries which are axial do not appear in the BDZ(I,J) array. Only diagonal boundaries appear in both arrays. This fact is useful in checking data accuracy using the boundary check program (to be discussed in a subsequent section).

The quantities NGX, NGY, VG, and GH, which are explained below, are read only if IFBDY#0. The IFBDY#0 option is for solving the potential distributions in a finer grid region (i.e., with a finer mesh) in the lower left-hand corner of a previously solved coarse mesh.

If IFBDY#0 the next data card contains NGX, NGY in format 2I4. NGX is the number of "coarse" axial grid lines in the "fine" grid solution region, including the axial boundary of the fine grid region (which must also be a "coarse" grid line) and the r=0 axis. The "coarse" grid lines are those that were used in the initial solution of the complete problem (i.e., with IFBDY=0) that was run before the IFBDY#0 option is to be executed. NGY is defined in an analogous manner for the coarse radial grid lines.

Next, the values GH(I) for I = 1 through NGX are read. These values are the coordinates of the r intercepts of the coarse axial grid lines. The format for GH(I) is 8F10.3.

The voltages VG(I), for I = 1 through NGX are read next. These are the node voltages along the radial coarse grid line which coincides with the top, radial grid line of the fine grid solution region. These voltages are obtained from the previously solved coarse grid solution. The format for VG(I) is 8F10.3.

In a like way, the coordinates are read for the z intercepts of the coarse radial grid lines, viz. GH(J), for J = 1 through NGY, format 8F10.3. Likewise, the voltages along the right side axial boundary of the fine grid, VG(J), for J = 1 through NGY are ready in format 8F10.3.

NGX, NGY, GH(1), VG(1), GH(J), VG(J) are the only optional data for the program. The following information is required in all cases, irrespective of the value of IFBDY.

NPOTS is read next, in format I4. NPOTS is the number of known potentials in the problem. For example, if part of the region has a voltage of +100 volts, and elsewhere there is a grounded region, with no other known potentials present, then NPOTS=2. The known potentials must always be greater than or equal to zero; no negative potentials are allowed. If negative potentials exist, the absolute value of the minimum potential must be added to all known potentials in the problem so that there are no negative potentials remaining.

Once NPOTS is read, the known potentials must be assigned to those nodes at which the potentials are known. This is done in a loop performed a number of times equal to NPOTS. The first information needed is POT, format F8.0, which is the potential assigned to a region (or several regions) of known potential. The "region" may also be a single point (i.e., a single node). Next, NLINES, format I4, is read. NLINES is the number of rectangles needed to describe all regions of the particular potential POT. These rectangles need not be connected. An example of how to calculate NLINES will be given with the aid of Figure 13. The fourteen dark dots are points of known potential. In this example, it takes two rectangles to describe the region of known potential. Note that one of the rectangles is "collapsed" to a line segment. A point would be considered as a rectangle that has "collapsed" to a point. Thus NLINES=2 in the example of Figure 13.

Next, the quantities L1, L2, M1, and M2 are read in format 4I4. L1 and L2 are the indices of the axial grid lines that form "edges" of a rectangle of potential POT. L1 must be less than or equal to L2. Likewise, M1 and M2 are the indices of the radial grid lines that form the remaining "edges" of the rectangle. M1 must be less than or equal to M2. So for the example of Figure 13, two groups of four numbers each are used to describe the location of the two rectangles. These numbers, read into the computer on two cards following the value of NLINES, would be 1-2,1-2,J-1,J followed by 1-1,1+2,J,J+2. As a further example, the values of L1, L2, M1, and M2 for a single point of known potential where the 22nd axial line and the 44th radial line intersect would be: 22 22 44 44. Each set of four values L1, L2, M1, and M2 must be on a separate data card.

The above procedure is repeated for each region (which may or may not be a "connected" region) of distinct potential POT, with the order of input data being: POT, NLINES, and NLINES groups of L1, L2, M1, M2.

This concludes the input data development for FVSOLVR; a summary of the order in which the input data cards are read in the FVSOLVR program is given below.

```

READ NX,NY,NEPR,NEPZ,IFSKIP,IFBDY
Format 6I4

READ (HX(I), I=1,NX)
Format 10F8.0

READ (HY(J), J=1,NY)
Format 10F8.0

READ ((BDR(I,J), J=1, NEPR), I=1,NY-1)
Format 20I4

READ KO,KN,KN
Format 3I4

READ (EPSR(K), K=1,KN)
Format 16F5.0

READ ((BDZ(I,J), J=1,NEPZ), I=1,NX-1)
Format 20I4

READ KO,KN,KN
Format 3I4

READ (EPSZ(K), K=1,KN)
Format 16F5.0

```

This series of data
 cards is repeated,
 as required (see the
 discussion on dielec-
 tric sequences).

This series of data
 cards is repeated,
 as required (see the
 discussion on dielec-
 tric sequences).

The following five READ statements are executed only if IFBDY \neq 0.

```

READ NGX,NGY
Format 2I4

READ (GH(I), I=1,NGX)
Format 8F10.3

READ (VG(I), I=1,NGX)
Format 8F10.3

READ (GH(J), J=1,NGY)
Format 8F10.3

READ (VG(J), J=1,NGY)
Format 8F10.3

READ NPOTS
Format I4

READ POT
Format F8.0

READ NLINES
Format I4

READ L1,L2,M1,M2
Format 4I4

```

Repeated NPOTS
 Times

Repeated NLINES Times

Checking Input Data

Once the data deck is set up, FVSOLVR can be executed, but it is advisable to check the data with three data check routines. This can save time and money by catching data errors before the solving routine is executed.

The first routine involves a boundary check program called BDYCHK. This program plots (using Calcomp plotting) dielectric boundaries in such a manner that a horizontal boundary appears as a horizontal line segment, at the given location, a vertical boundary line appears as a vertical line segment, and a diagonal boundary appears as two intersecting slashes forming an X. All diagonal boundaries appear in both the BDR(I,J) data and the BDZ(I,J) data. Therefore the appearance of only one slash of the X indicates an error in the BDR(I,J) and/or the BDZ(I,J) data, since one slash (\) of the X comes from the BDR(I,J) data and the other slash (/) of the X comes from the BDZ(I,J) data. An example of such a plot for the configuration of Figure 8 is shown in Figure 14. Numerical tables are also printed by the boundary check program to aid location of improper data.

The data deck set up for the boundary check program BDYCHK is as follows. The first card contains NX, NY, NEPR, and NEPZ. These integer variables are defined in the same way as in FVSOLVR. The second card contains IO, IN, JO, and JN. These data outline the region to be examined by the boundary check program. IO and IN are the indices of the axial grid lines and JO and JN are the indices of the radial grid lines that form the borders (or edges) of the region which the boundary check program is to examine. IO is less than IN and JO is less than JN. The HX(I) cards come next, followed by the HY(J) cards, the BDR(I,J) cards and, lastly, the BDZ(I,J) cards. The cards containing the HX(I), HY(J), BDR(I,J), and the BDZ(I,J) data are the exact same cards as those prepared for the FVSOLVR program. In summary, the deck set up is as follows.

```
READ NX,NY,NEPR,NEPZ
Format 4I4

READ IO,IN,JO,JN
Format 4I4

READ (HX(I), I=1,NX)
Format 10F8.0

READ (HY(J), J=1,NY)
Format 10F8.0

READ ((BDR(I,J), J=1,NEPR), I=1,NY-1)
Format 20I4

READ ((BDZ(I,J), J=1,NEPZ), I=1,NX-1)
Format 20I4
```

Once the boundary data are verified, a check of the dielectric constant information is performed. The given dielectric constants must be assigned within the boundaries. The name of the program which performs this check is EPSCHK. The EPSCHK program uses the boundary information, BDR(I,J), BDZ(I,J), and the dielectric constant assignment information, KO, KN, KK, and EPSR(KK) for the radial direction and KO, KN, KK and EPSZ(KK) for the axial direction, in order to generate (as printed output) two arrays of values of dielectric constants. One array contains the values of dielectric constants encountered in the radial direction search (rows), and the other array contains the values of dielectric constants encountered in the axial direction search (columns). Two tables are printed out, one for each array of dielectric constants. The two arrays are compared and inconsistencies between them are noted in the program output.

The data deck set up for the EPSCHK program is as follows. The first card contains NX, NY, NEPR, and NEPZ; this is the same as the NX, NY, NEPR, NEPZ card used in the boundary check program. The BDR(I,J) come next, followed by the KO, KN, KK, EPSR(KK) cards, set up in the same way as shown in the FVSOLVR program. Finally, the BDZ(I,J) cards and accompanying KO, KN, KK, EPSZ(KK) cards are read. A summary of the order of reading data in the EPSCHK program is as follows.

```

READ NX,NY,NEPR,NEPZ
Format 4I4

READ ((BDR(I,J), J=1,NEPR), I=1,NY-1)
Format 20I4

READ KO,KN,KK
Format 3I4
READ (EPSR(K), K=1,KK)
Format 16F5.0
] Repeated, as required.

READ ((BDZ(I,J), J=1,NEPZ), I=1,NX-1)
Format 20I4

READ KO,KN,KK
Format 3I4
READ (EPSZ(K), K=1,KK)
Format 16F5.0
] Repeated, as required.

```

A third data check is used to verify the values and locations of known input potentials. This check involves the use of FVSOLVR itself. The check is implemented by simply using the data deck to be used with the actual program, except with IFSKIP \neq 0 in the NX, NY, NEPR, NEPZ, IFSKIP, IFBDY card. This results in FVSOLVR skipping execution of the solving algorithm and printing out all known potentials at their given grid locations and "-1" at all other grid locations

Output From FVSOLVR

Potentials are printed by a line printer as well as written on tape; therefore the user must supply the tape. Of course, if the user wishes to change the output statements in the program, the output may be obtained in any form. The array $V(I,J)$ contains the potential solution at the nodes having coordinates $(HX(I),HY(J))$.

Potential listings for FVSOLVR are printed out in the following manner. Blocks of 50 values of potential in the axial direction and 14 values in the radial direction, maximum, are printed on each page of output from the computer. The table is started from the lower left hand corner of the solution region and proceeds to the right producing blocks of 14×50 potential values. This is illustrated in Figure 15. If, for example, as in "page 3" in Figure 15, fewer than 14 values of potential are needed to complete the printing of the data in the radial direction, then the printout will fill only a portion of the page. A similar procedure is followed in the axial direction, until the entire solution region is printed.

USE OF THE PLOT PROGRAM FVPLLOT

Use of the plotter program FVPLLOT will now be described. The plotter can be set up to plot equipotential and/or flux lines. However, as a result of the inherent nature of the plotting program, flux lines can be plotted only for problems of the "point gap" type. Examples of "point gap" problems are shown in Figure 16. Plotting flux lines in configurations other than the "point gap" type may cause difficulties. Equipotential lines can always be plotted, irrespective of whether or not the problem is of a "point gap" nature.

Data for FVPLLOT may be input from cards or tape. The user must use the appropriate "read" statements. The first information read is IVU. If $IVU=1$, the program plots only the potential lines. If $IVU=2$, the program plots only the field lines. If $IVU=3$, the program plots both potential and field lines. (If flux lines are to be plotted, FVSOLVR must be executed to solve for the necessary field values. For details, see CEL Technical Note N-1502.)

Next, a card containing IO, IN, JO, JN, NQUAD, KK, VMIN, and SZ is read. IO and IN are the first and last indices, respectively, of the axial grid lines that form edges of the region to be plotted. JO and JN are the first and last indices, respectively, of the radial grid lines that form edges of the region to be plotted. NQUAD indicates whether only the solution region solved for by FVSOLVR is to be plotted or whether "mirror images" of this solution region are also to be plotted, as is illustrated in Figure 17. If $NQUAD=1$, only the solution region obtained from FVSOLVR is plotted; if $NQUAD=2$, one mirror reflection is introduced; if $NQUAD=4$, two mirror reflections are introduced. This feature enables problems of varying degrees of symmetry to be solved by

FVSOLVR and then plotted in their entirety by FVPLLOT while not necessarily inputting the full geometrical configuration as data to either program. KK is the number of equipotential lines to be plotted "automatically" by FVPLLOT. SZ is the voltage increment between successive equipotential lines and VMIN is the potential value which is incremented by SZ to determine the first equipotential line to be plotted. Successive increments of SZ are added and an equipotential line is plotted for each of the KK potential values. The first potential value plotted is $VMIN + SZ$ and the last value plotted is $VMIN + (KK \times SZ)$. The value VMIN is not plotted because it is assumed that VMIN usually represents an electrode or a boundary of the solution region. Likewise, any known voltages, such as electrodes, should not be plotted "automatically" as just described. These voltages should be given as separate input data to FVGRPHR, which will be explained later. As an example, if two electrodes of known potential are 0.0 and 100.0 volts and equipotential lines are desired at 10.0 volt intervals, then $KK=9$, $VMIN=0.0$, and $SZ=10.0$. The 0.0 and 100.0 volt lines are not plotted "automatically" and are not counted when determining KK.

NX and NY are read next. NX is the number of axial grid lines and NY is the number of radial grid lines used in the FVSOLVR solution. Following this, the HX(I) and HY(J) data are read in the order HX(1), HX(2), HX(3), ..., HX(NX), HY(1), HY(2), HY(3), ..., HY(NY). Then the voltages V(I,J), previously calculated (by FVSOLVR) at the grid points, are read as input to FVPLLOT. The NX, NY, HX(I), HY(J), and V(I,J) data can be stored on tape directly from the FVSOLVR solution and read from the same tape into FVPLLOT.

If $IVU \neq 1$, separate data must be input for the field (flux lines) plot. If $IVU=1$, the separate field data is to be omitted. The field data are MX and MY (which are exactly analogous to the number of grid lines, NX and NY, in the equipotential case), GX(I), GY(J), and U(I,J) (which are exactly analogous to HX(I), HY(J), and V(I,J) for the equipotential case). This information can also be stored on tape from the FVSOLVR solution and read from the same tape as input to FVPLLOT.

The last information inputted is that of various reference lines into the plot. Such lines may include known potential lines (as previously mentioned), dielectric boundaries, or any other lines that the user wishes to plot. Data needed to plot known potential or reference lines is as follows. NLINES is the number of lines drawn by the plotter to generate all figures and lines. In determining the numerical value of NLINES, a distinct line is counted each time the pen is lifted off the paper. NPTS, which is read next, is the number of points needed to generate a given line. For each line that is to be plotted, a data card containing a value for NPTS is followed by a card (or cards) containing values of LX(N) and LY(N), with the LX(N) and LY(N) data appearing on the cards in the order LX(1), LY(1), LX(2), LY(2), LX(3), LY(3), ..., LX(NPTS), LY(NPTS). NPTS is the number of points being used to generate the particular line, and LX(N) and LY(N) are the indices of the axial and radial grid lines, respectively, which intersect at point N. The

plotter draws straight lines between these points. If a long straight line is to be drawn, only the end points of the line need be given. Each time the line turns a new point is needed.

The following is a short review of the order in which the input data cards are read in the FVPLOT program. Each READ statement by which data are read most conveniently from tapes is indicated by an asterisk.

```
READ IVU
Format I4

READ IO,IN,JO,JN,NQUAD,KK,VMIN,SZ
Format 6I4, 2F5.0

*READ NX,NY
Format 2I4

*READ (HX(I), I=1, NX), (HY(J), J=1,NY)
Format 8F10.3

*READ ((V(I,J), I=1,NX), J=1,NY)
Format 8F10.3
```

The previous five READ statements are executed irrespective of the value assigned to the integer variable IVU. The following three READ statements are executed only if IVU=2 or IVU=3.

```
*READ MX,MY
Format 2I4

*READ (GX(I), I=1,MX), (GY(J), J=1,MY)
Format 8F10.3

*READ ((U(I,J), I=1,MX), J=1,MY)
Format 8F10.3

READ NLines
Format I4

READ NPTS
Format I4

READ (LX(N), LY(N), N=1,NPTS)      Repeat NLines times
Format 20I4
```

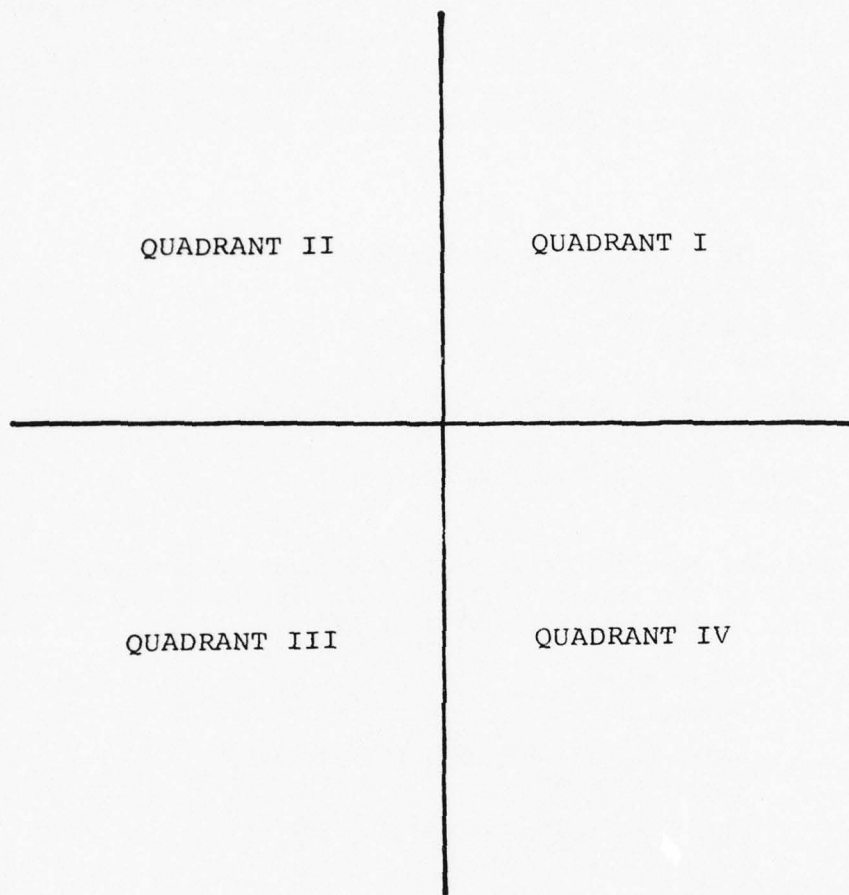


Figure 1. Labeling of quadrants in the cartesian coordinate system.

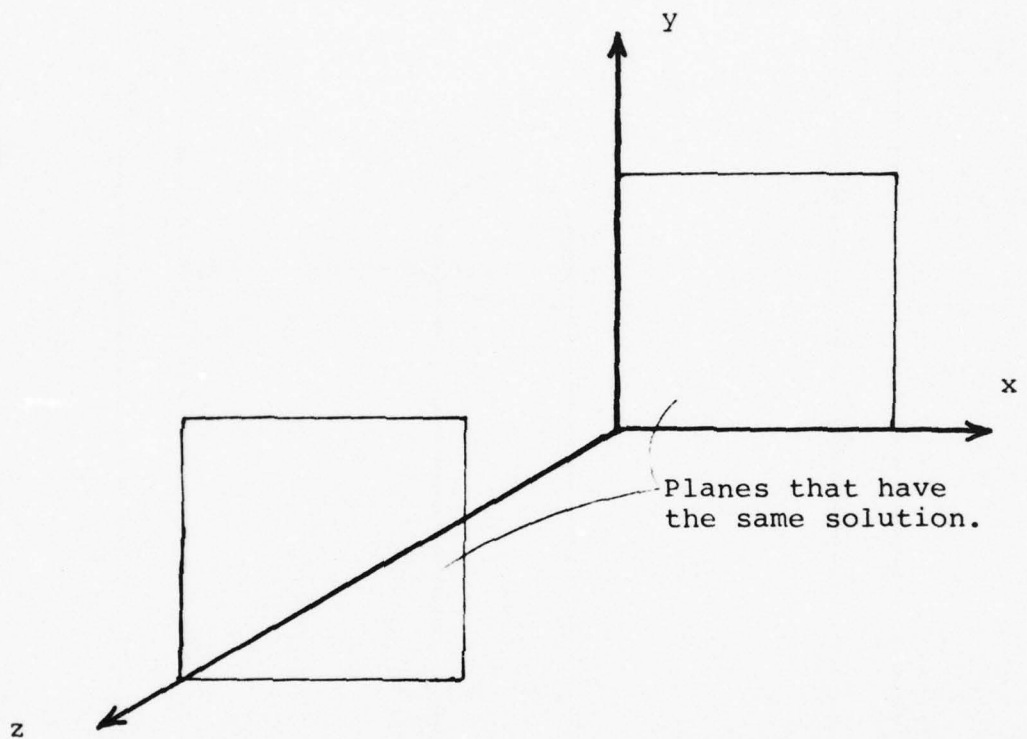
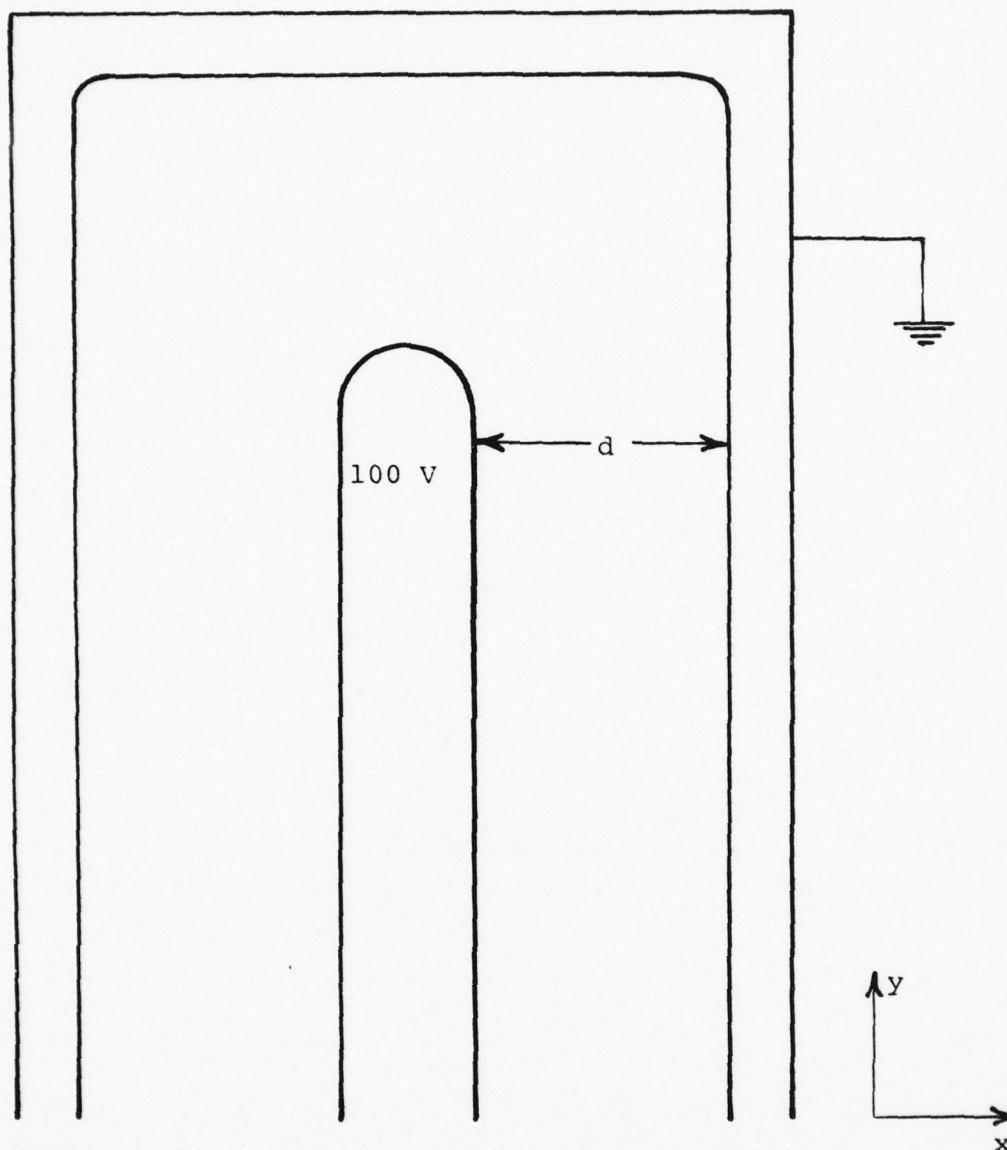


Figure 2. Illustration of two planes having identical solutions in the case of the cartesian coordinate system



Capacitor plate within a housing. The geometry of the capacitor and the field solution is the same in any plane perpendicular to the z -axis (the z -axis is perpendicular to the plane of the paper).

Figure 3. Example of a problem described in a cartesian coordinate system.

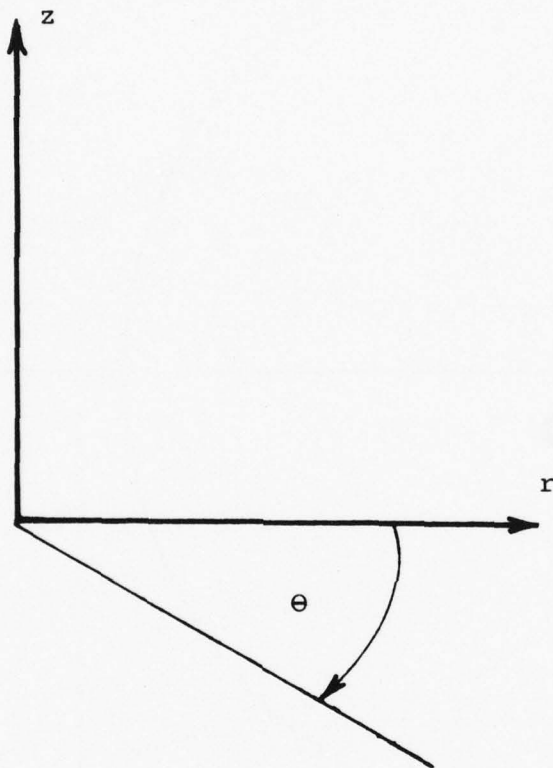


Figure 4. Coordinates used in the cylindrical system.

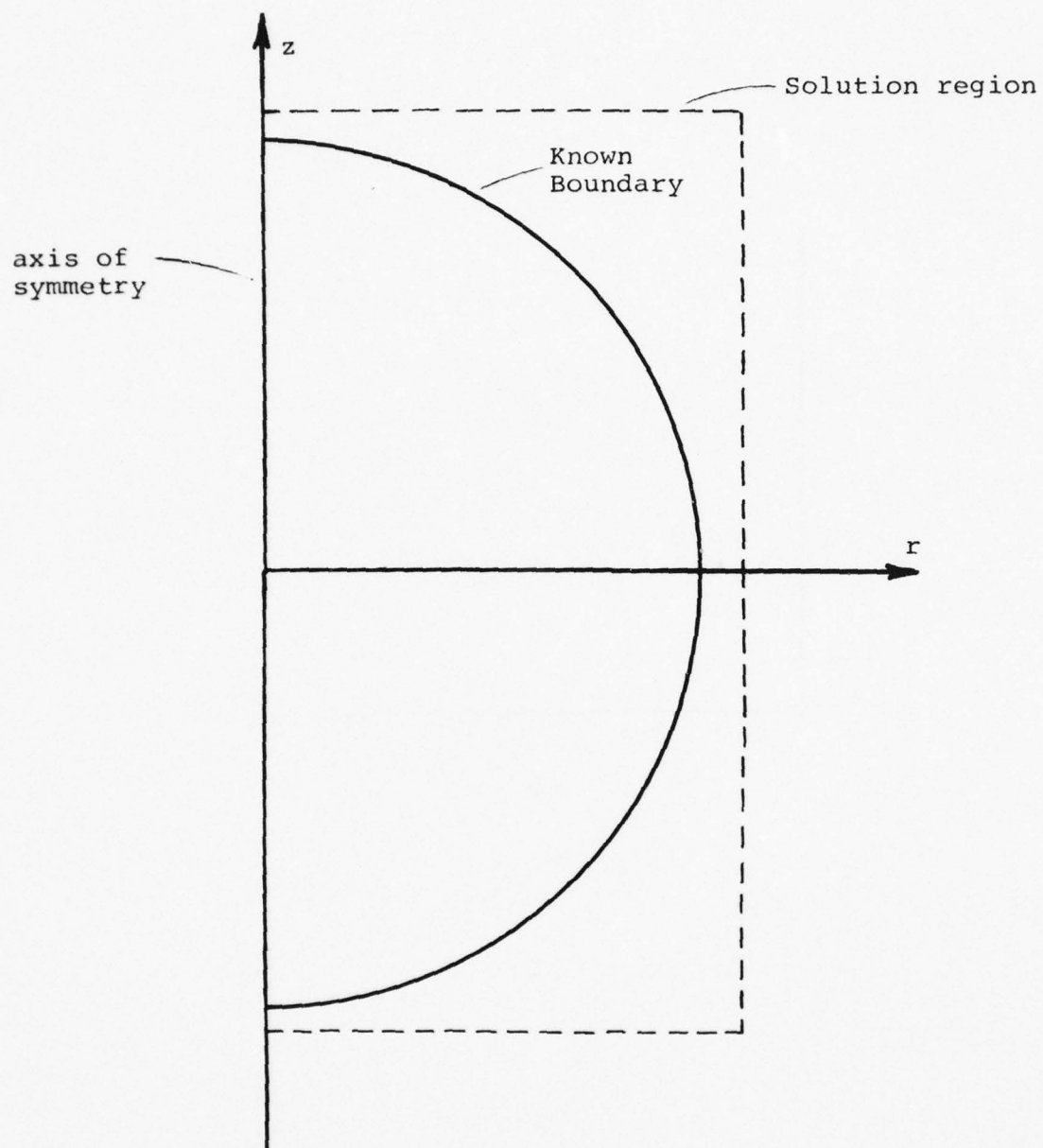


Figure 5. Illustration of a case of a circular known boundary within the solution region in the cylindrical system.

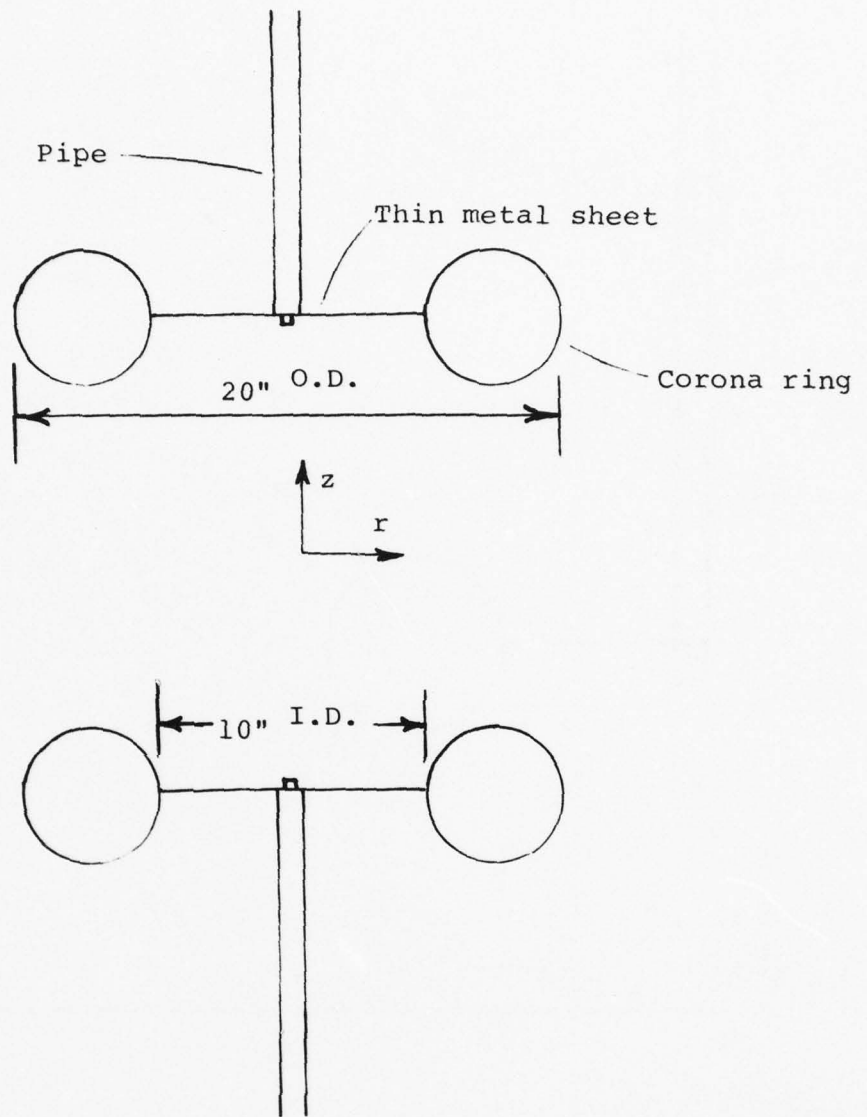


Figure 6. Geometry of corona rings suspended by pipes.

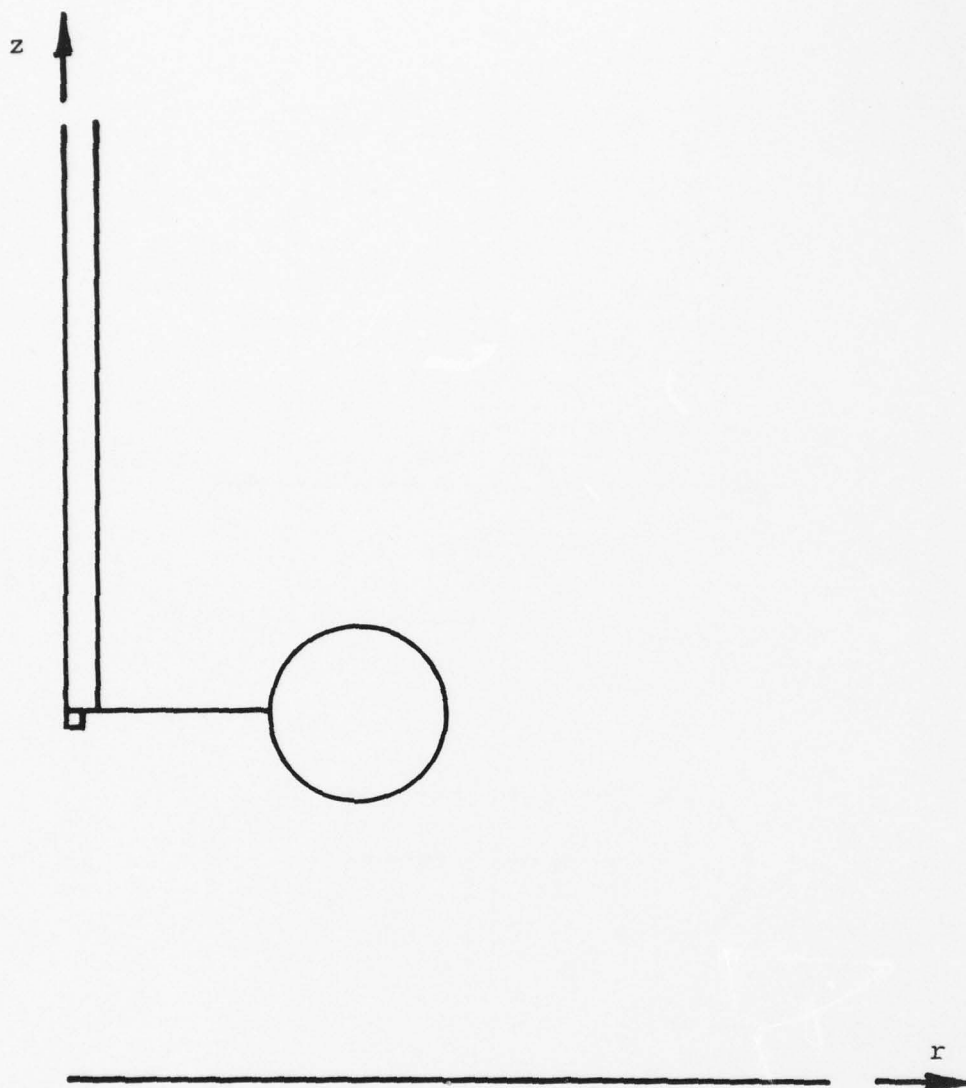


Figure 7. The reduced solution region for the corona rings shown in Figure 6 after taking symmetry into account.

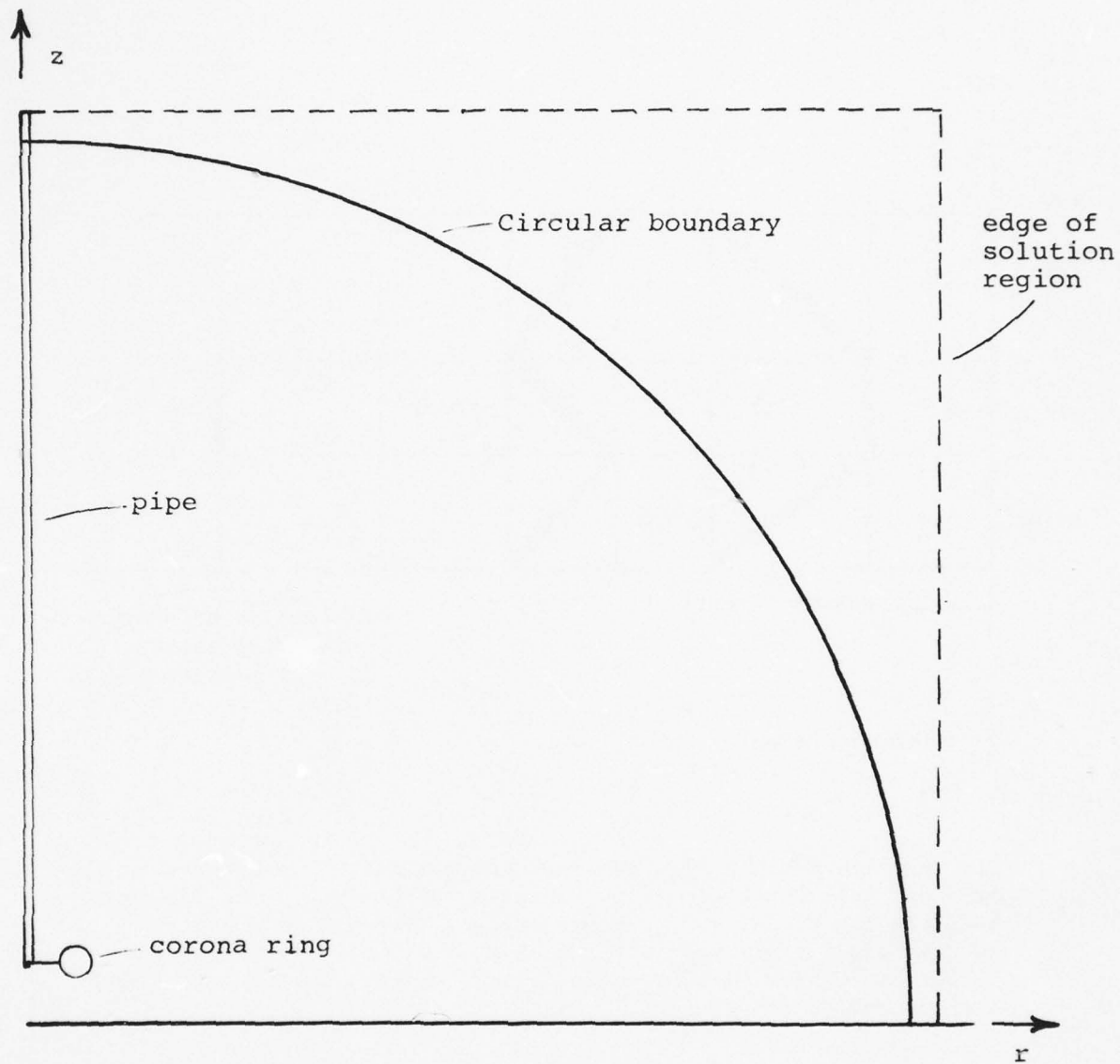
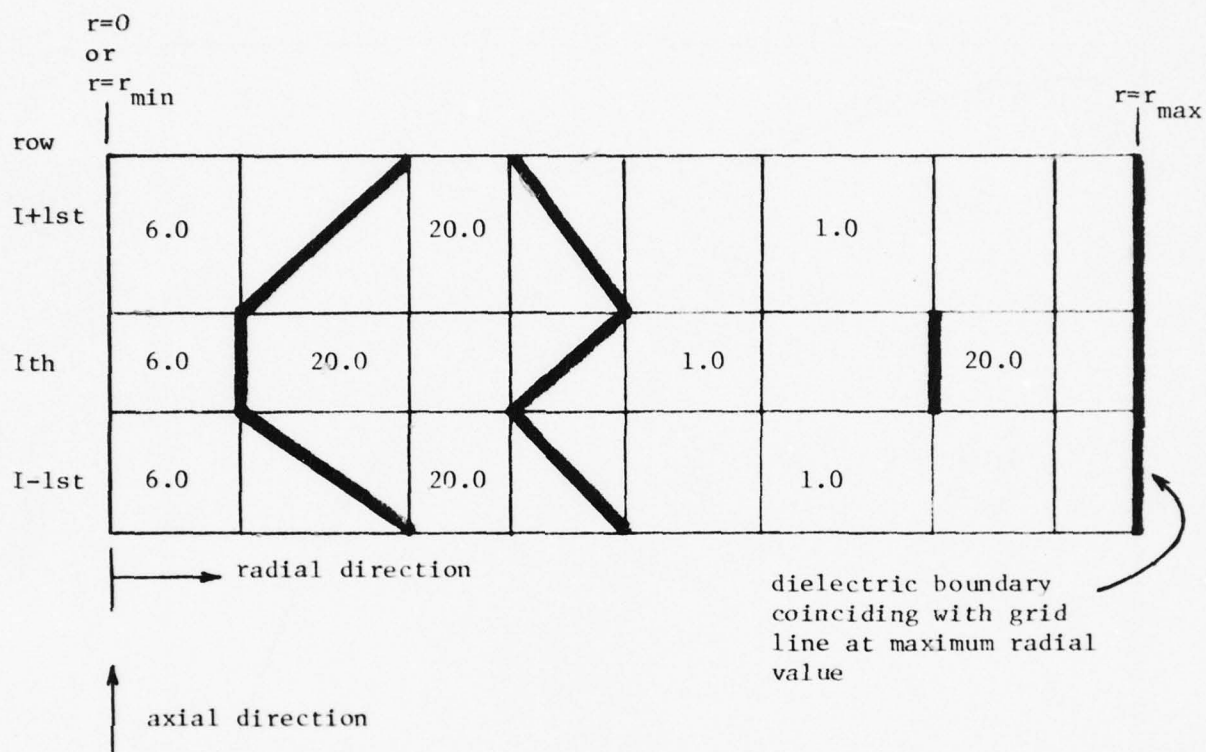
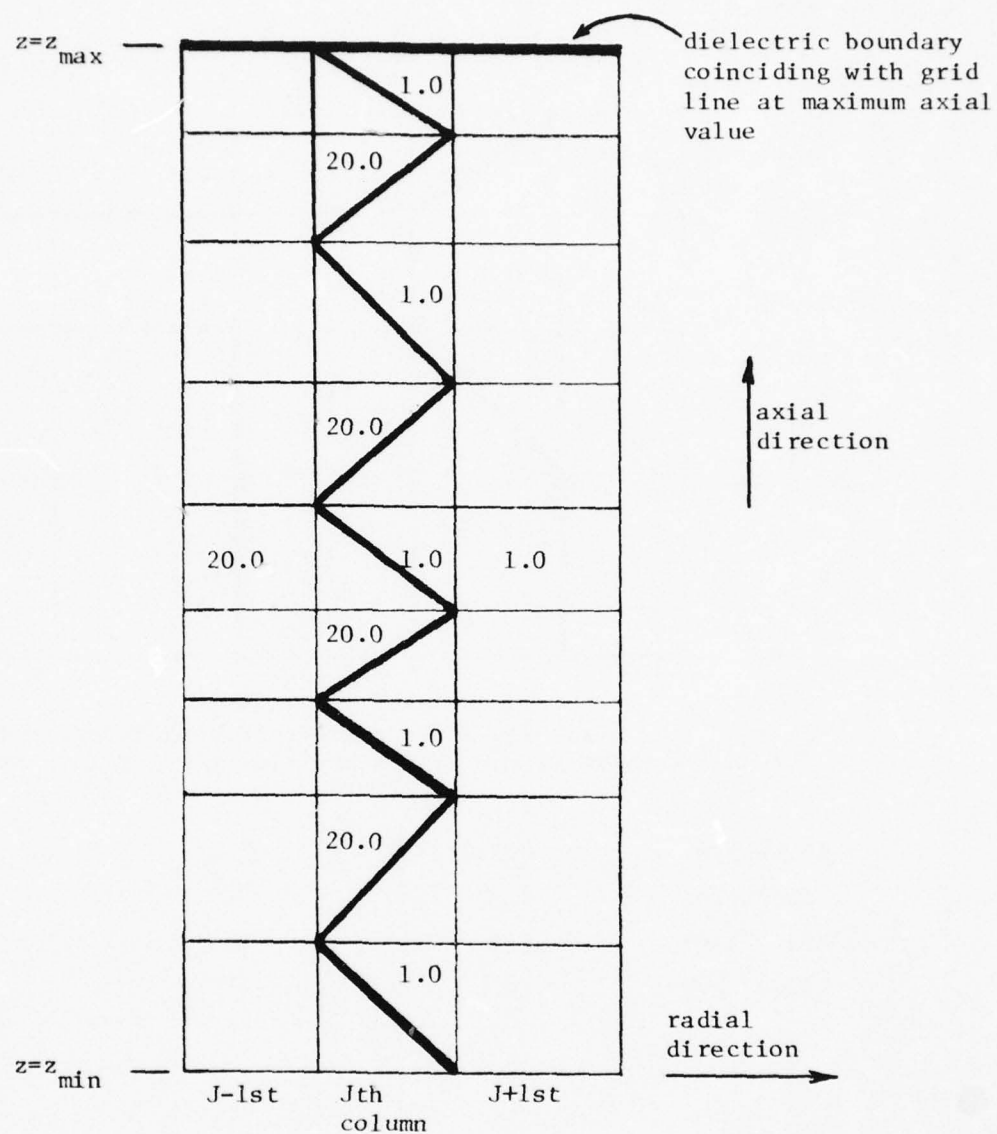


Figure 8. Reduced solution region with circular boundary



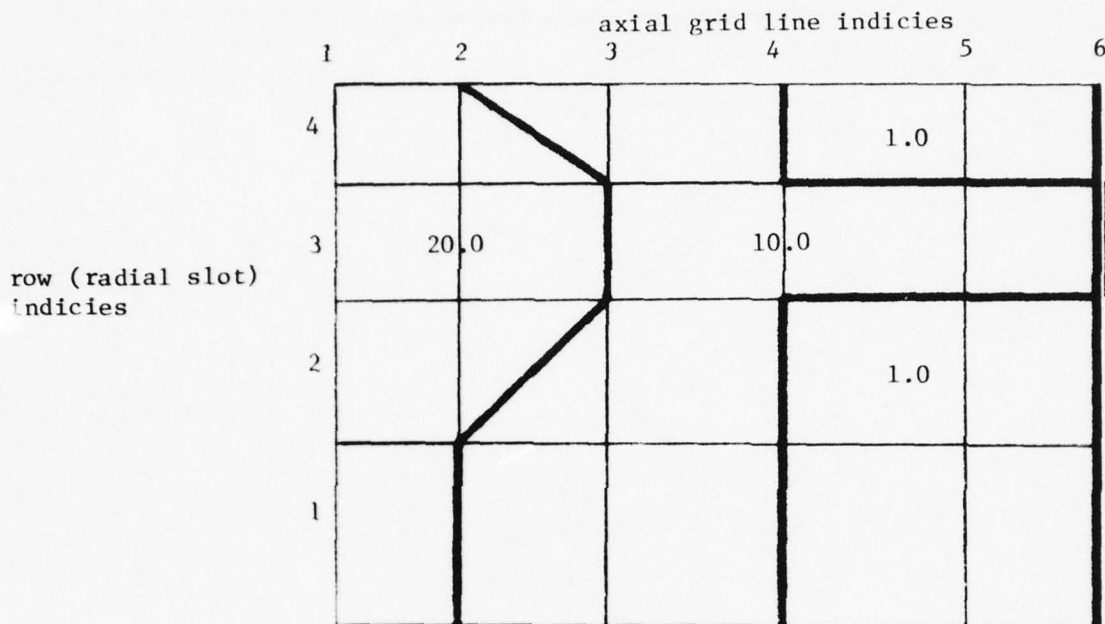
Dielectric boundaries are shown as heavy lines. The numbers in the grid are relative dielectric constants. That is, in the I th row, the relative dielectric constant changes from 6.0 to 20.0 to 1.0 to 20.0 in passing outward from the axis of symmetry in the radial direction. There must be a dielectric boundary on the grid line of maximum radial value.

Figure 9. Examples of dielectric boundaries encountered in the radial direction.



Dielectric boundaries are shown as heavy lines. The numbers in the grid are relative dielectric constants. That is, in the Jth column, the relative dielectric constant changes from 20.0 to 1.0 to 20.0 to ...1.0 to 20.0 to 1.0 in going from bottom to top in the axial direction. There must be a dielectric boundary on the grid line of maximum axial value.

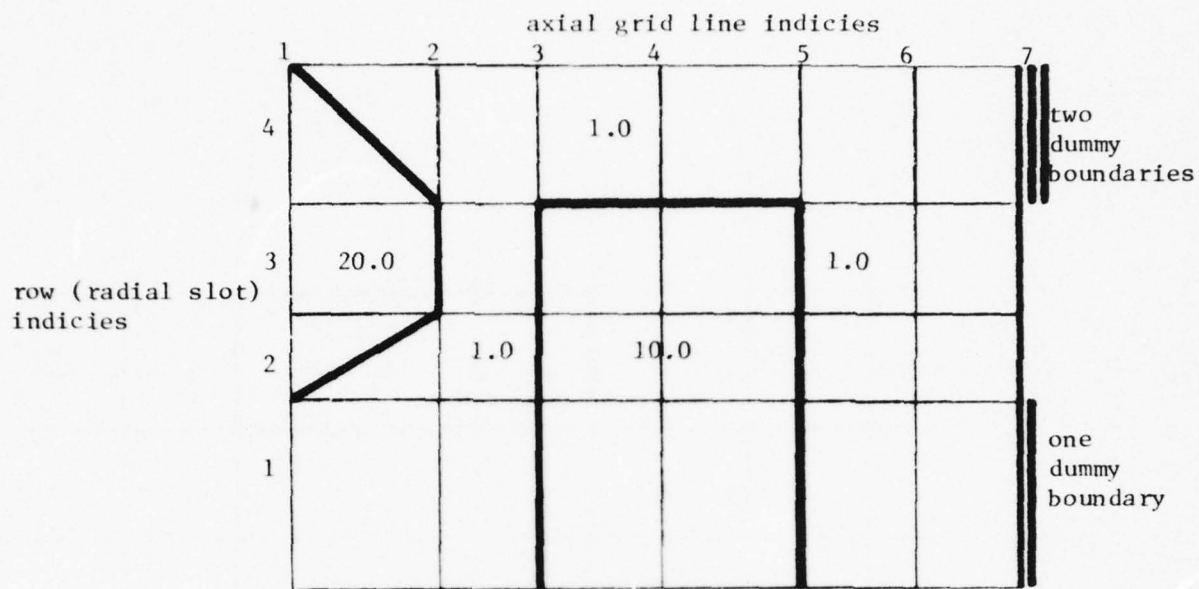
Figure 10. Examples of dielectric boundaries encountered in the axial direction.



The heavy lines are dielectric boundaries; and the numbers in the grid are dielectric constant values.

BDR(1,1)=4	BDR(1,2)=8	BDR(1,3)=12
BDR(2,1)=5	BDR(2,2)=8	BDR(2,3)=12
BDR(3,1)=6	BDR(3,2)=12	BDR(3,3)=12
BDR(4,1)=5	BDR(4,2)=8	BDR(4,3)=12

Figure 11. Numbering scheme used in the assignment of the BDR (I,J) values where radially encountered dielectric boundaries are specified.



The heavy lines are dielectric boundaries; and the numbers in the grid are dielectric constant values

$BDR(1,1)=6$	$BDR(1,2)=10$	$BDR(1,3)=14$	$BDR(1,4)=14$
$BDR(2,1)=3$	$BDR(2,2)=6$	$BDR(2,3)=10$	$BDR(2,4)=14$
$BDR(3,1)=4$	$BDR(3,2)=6$	$BDR(3,3)=10$	$BDR(3,4)=14$
$BDR(4,1)=3$	$BDR(4,2)=14$	$BDR(4,3)=14$	$BDR(4,4)=14$

Figure 12. Example of the use of several dummy boundaries at the last axial grid line to complete the $BDR(I,J)$ array.

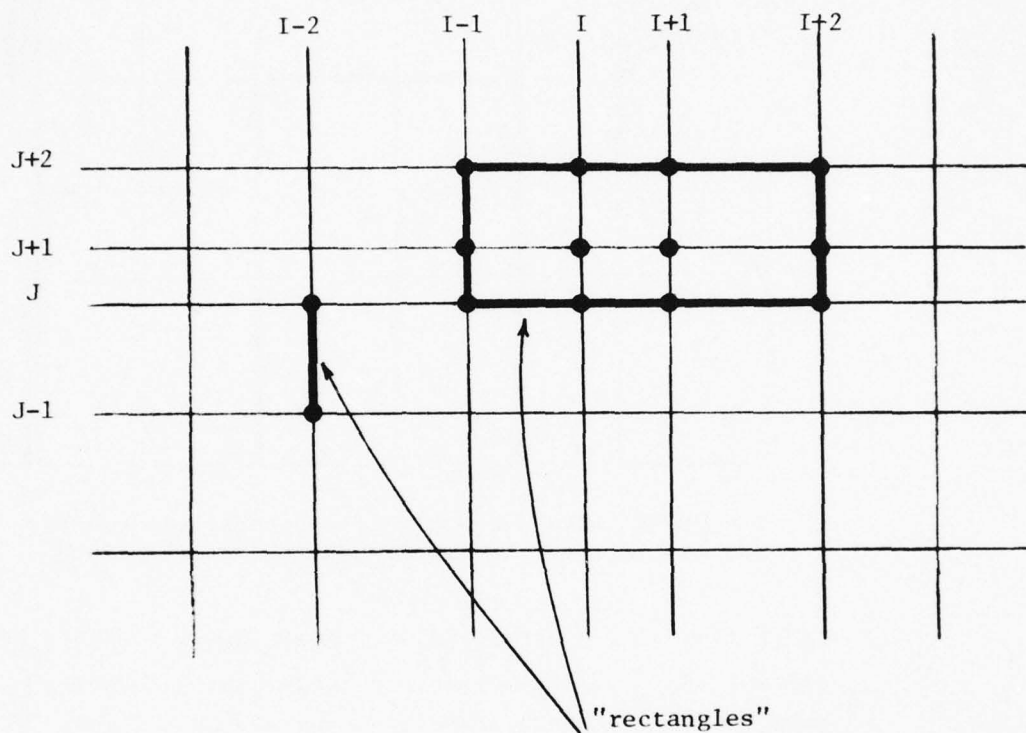


Figure 13. A region of known potential "POT" which is described by two "rectangles".

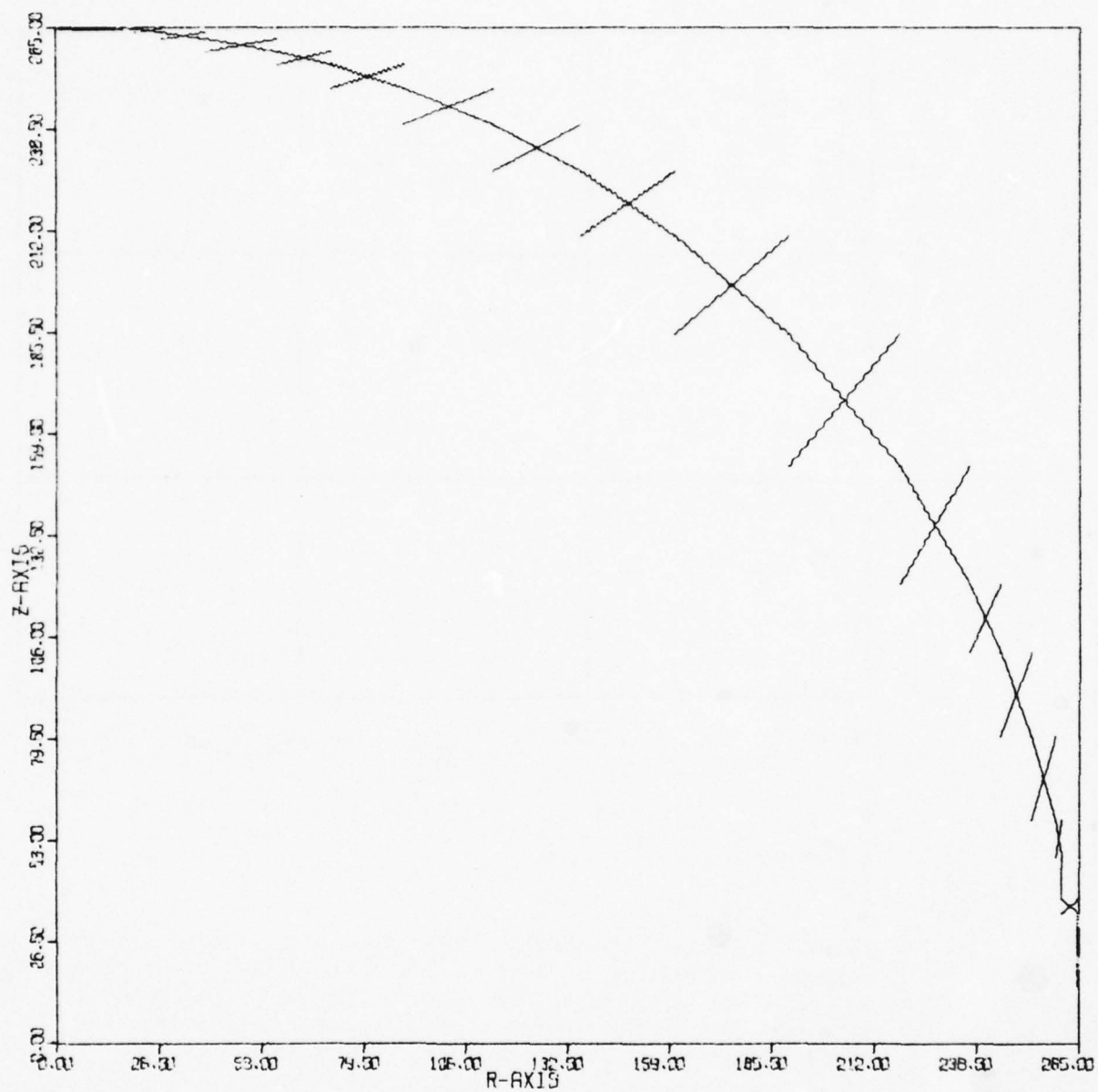
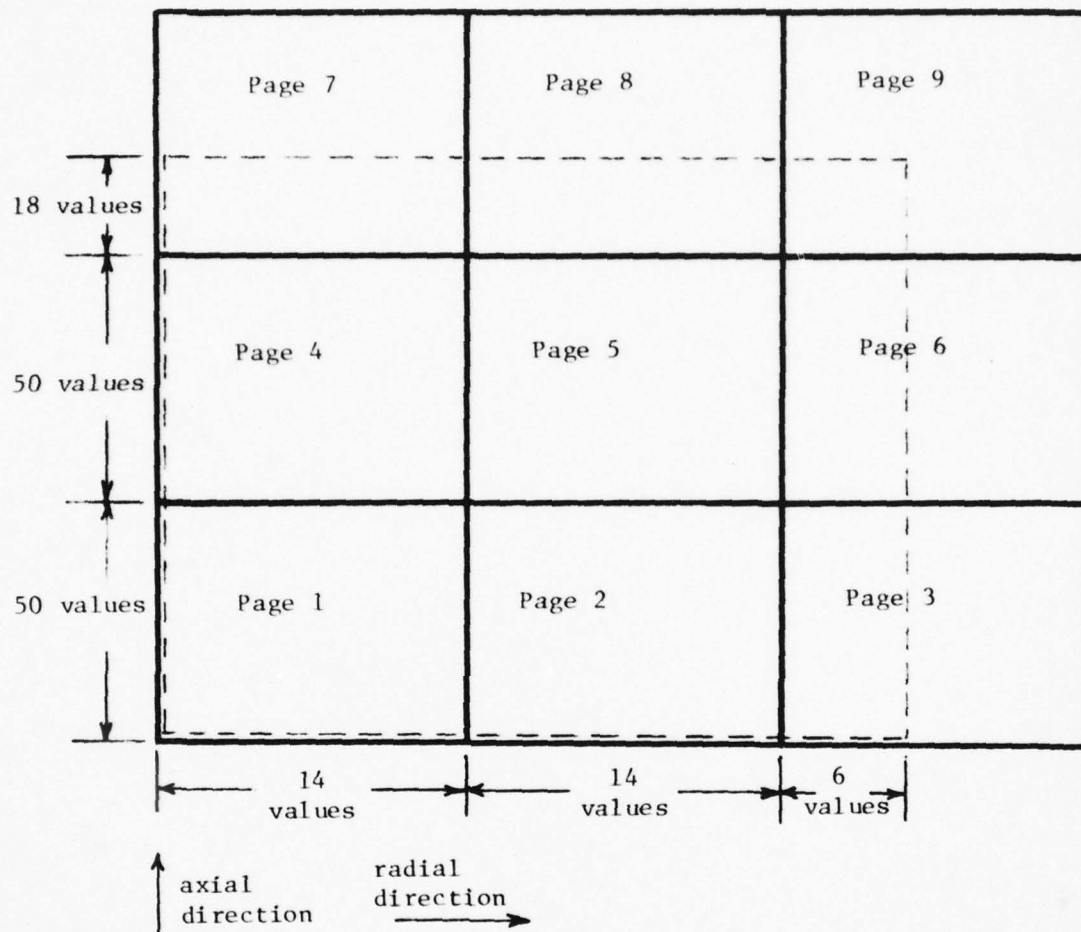


Figure 14. An example of a plot from BDYCHK, the boundary check program, for the configuration of Figure 8.



In this example $NX=34$ and $NY=118$. There exists correspondence between the position of data in the printout and the location in the solution region. The solution region is indicated by a dashed line in this figure. In this particular case nine pages of computer printed output are required.

Figure 15. Example of the format of the printout of field values from FVSOLVR.

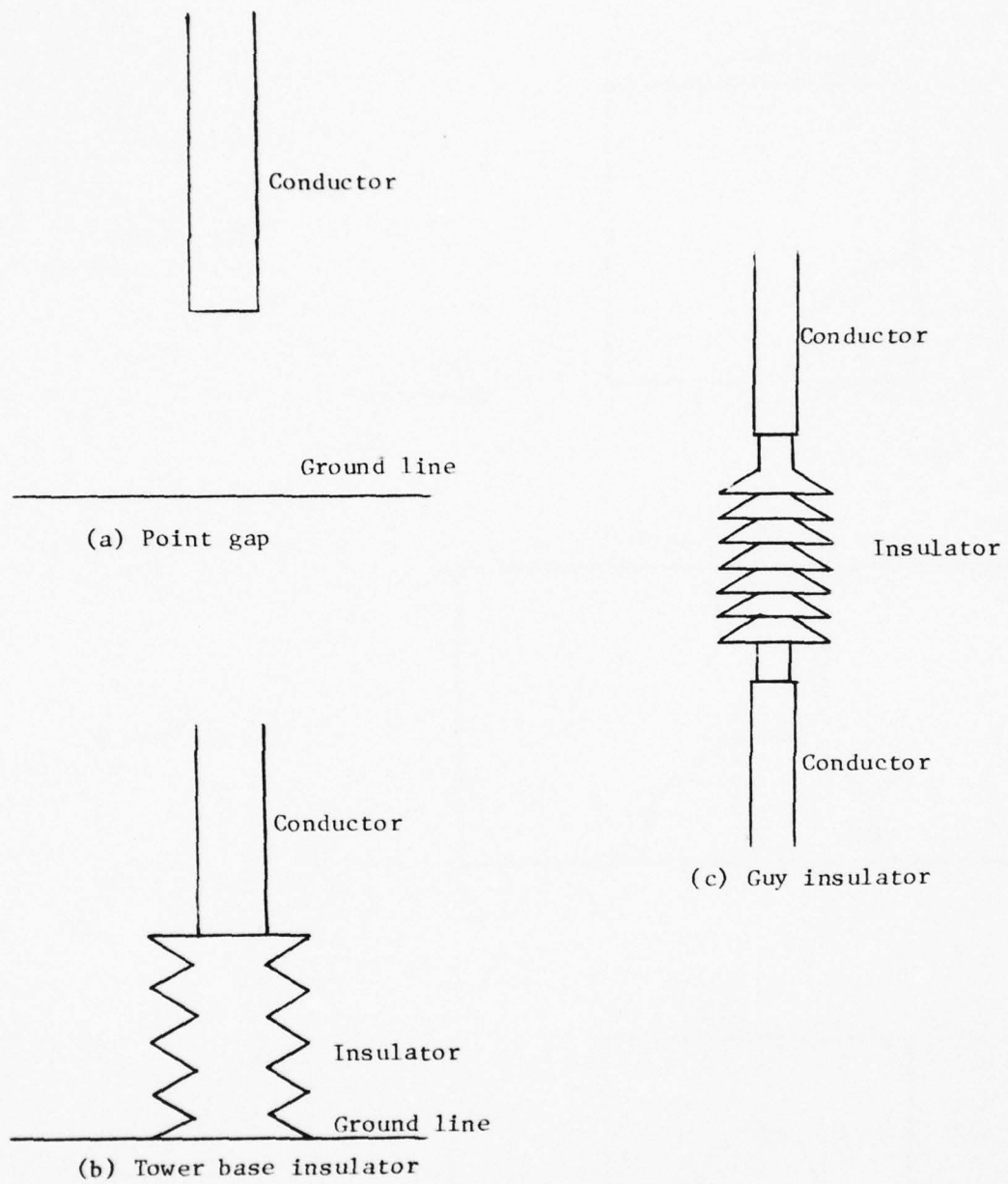
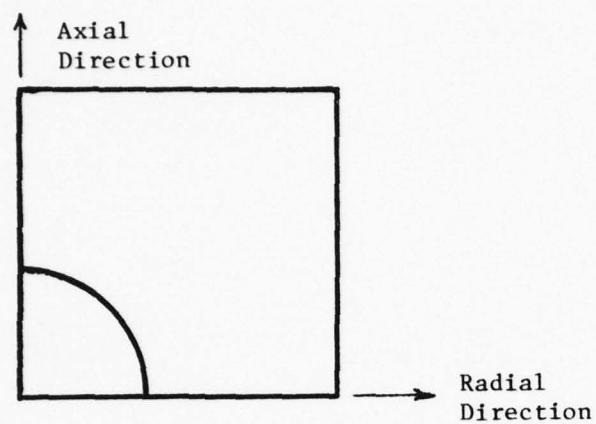
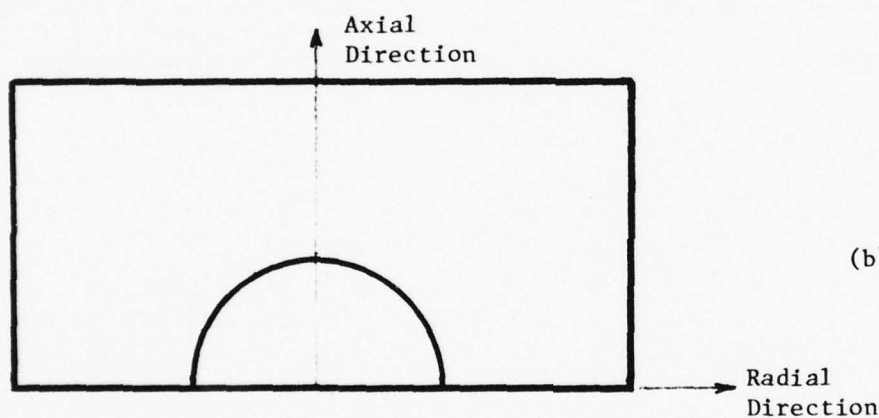


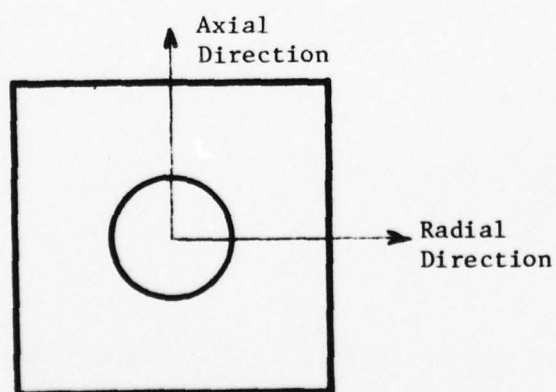
Figure 16. Examples of problems of a point gap nature.



(a) NQUAD=1
1 Quadrant Plotted



(b) NQUAD=2
2 Quadrants Plotted



(c) NQUAD=4
4 Quadrants Plotted

Figure 17. Illustration of the effect of the NQUAD assignment statement.

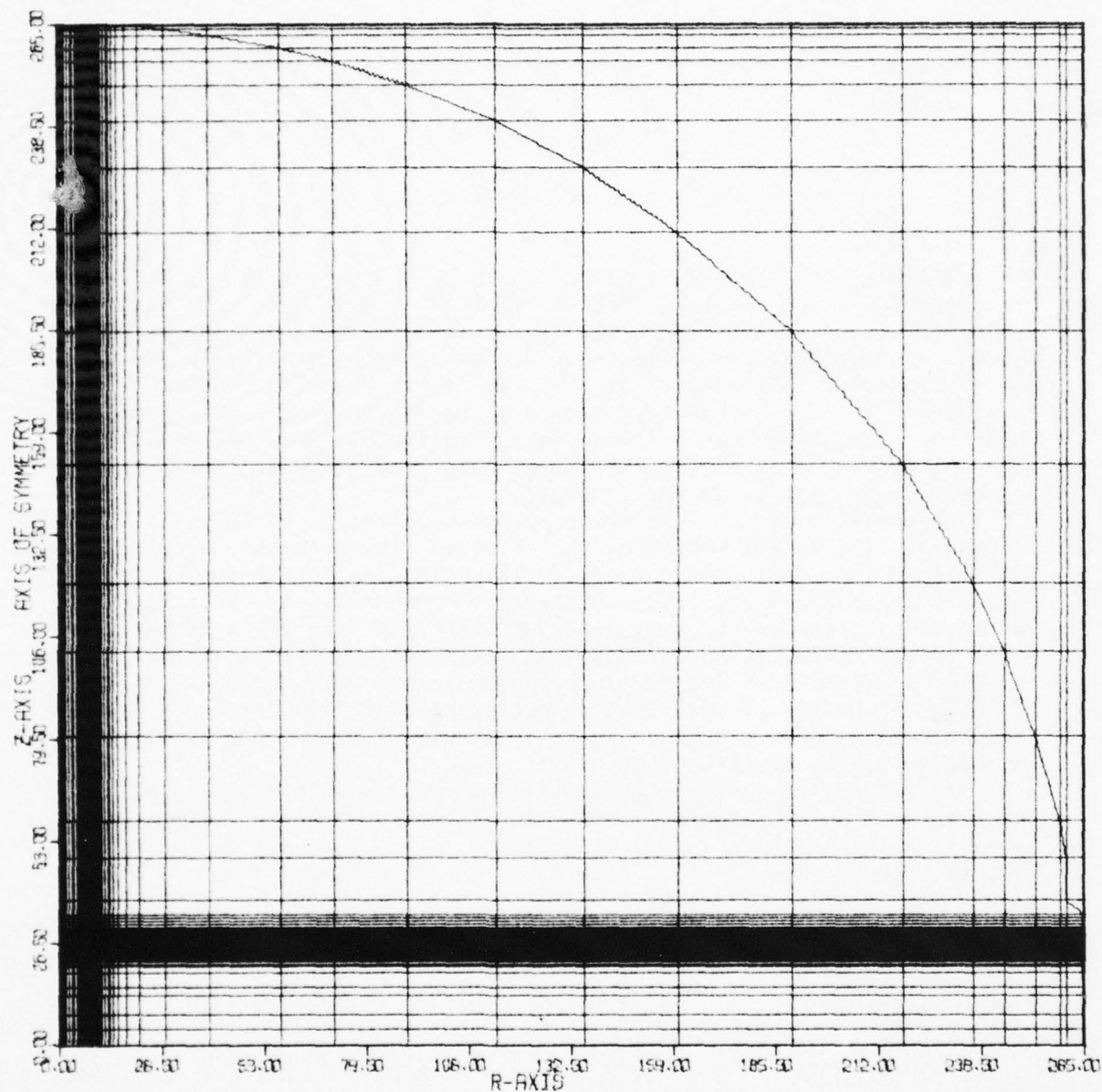
Appendix A

EXAMPLE PROBLEM

An example of a potential distribution problem is given to illustrate the organization of the input data for FVSOLVR and FVPLT. This simple problem, which is described in the text and illustrated in Figures 6, 7, and 8, involves only two dielectrics and two given potentials. One of the dielectric values is zero and is used only to satisfy boundary conditions near the top and right-hand edges of the grid. Many problems which are commonly encountered are more complicated than this example, and will require a more elaborate input data set. However, the data format will be just as in this example.

The corona ring and pipe are assigned a potential of 100 volts. The radial line at the bottom of the solution region (plane of symmetry) is assigned a potential of 0 (zero) volts. The region inside the circular boundary is assigned a relative dielectric constant of 1. This includes the regions of the corona ring and pipe which can have any arbitrary dielectric constant since the potential in those regions is known. The value of 1 is chosen for convenience. The region outside the circular boundary is assigned a relative dielectric constant of 0 (zero). Thus it is not necessary to assign boundary potentials to the top and right-hand edges of the solution region.

The grid lines in the solution region are shown in Figures A-1 and A-2. Table A-1 lists the input data for FVSOLVR. Table A-2 lists the input data for FVPLT for the equipotential line plot near the corona ring as shown in Figure A-3. Figure A-4 is an equipotential line plot of the entire solution region.



The grid lines near the corona ring are too densely packed to allow good resolution in this figure. Figure A-2 illustrates the grid lines near the corona ring.

Figure A-1. Grid lines for the entire solution region for the corona ring problem shown in Figure 8.

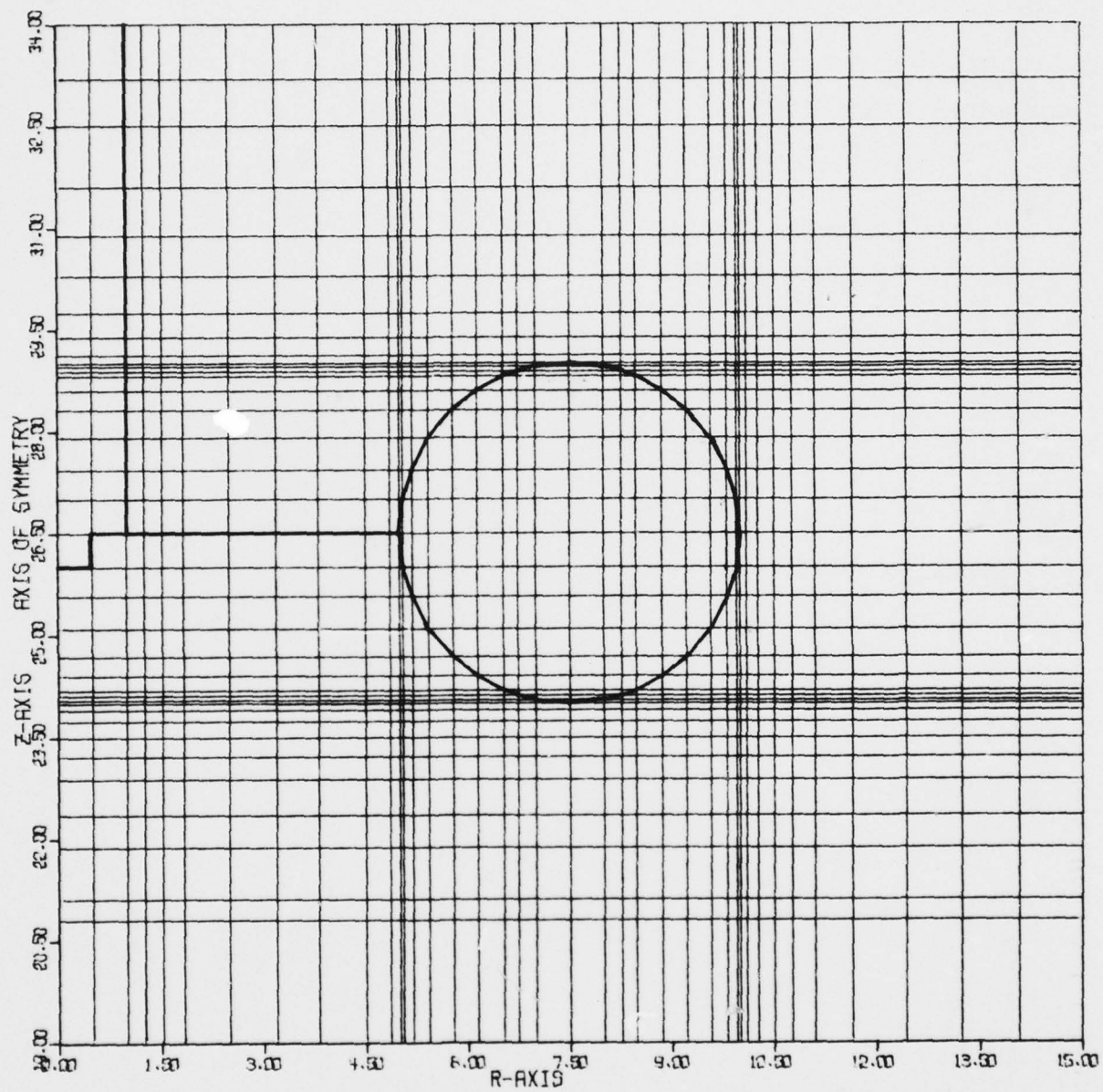


Figure A-2. Illustration of the grid lines in the neighborhood of the corona ring.

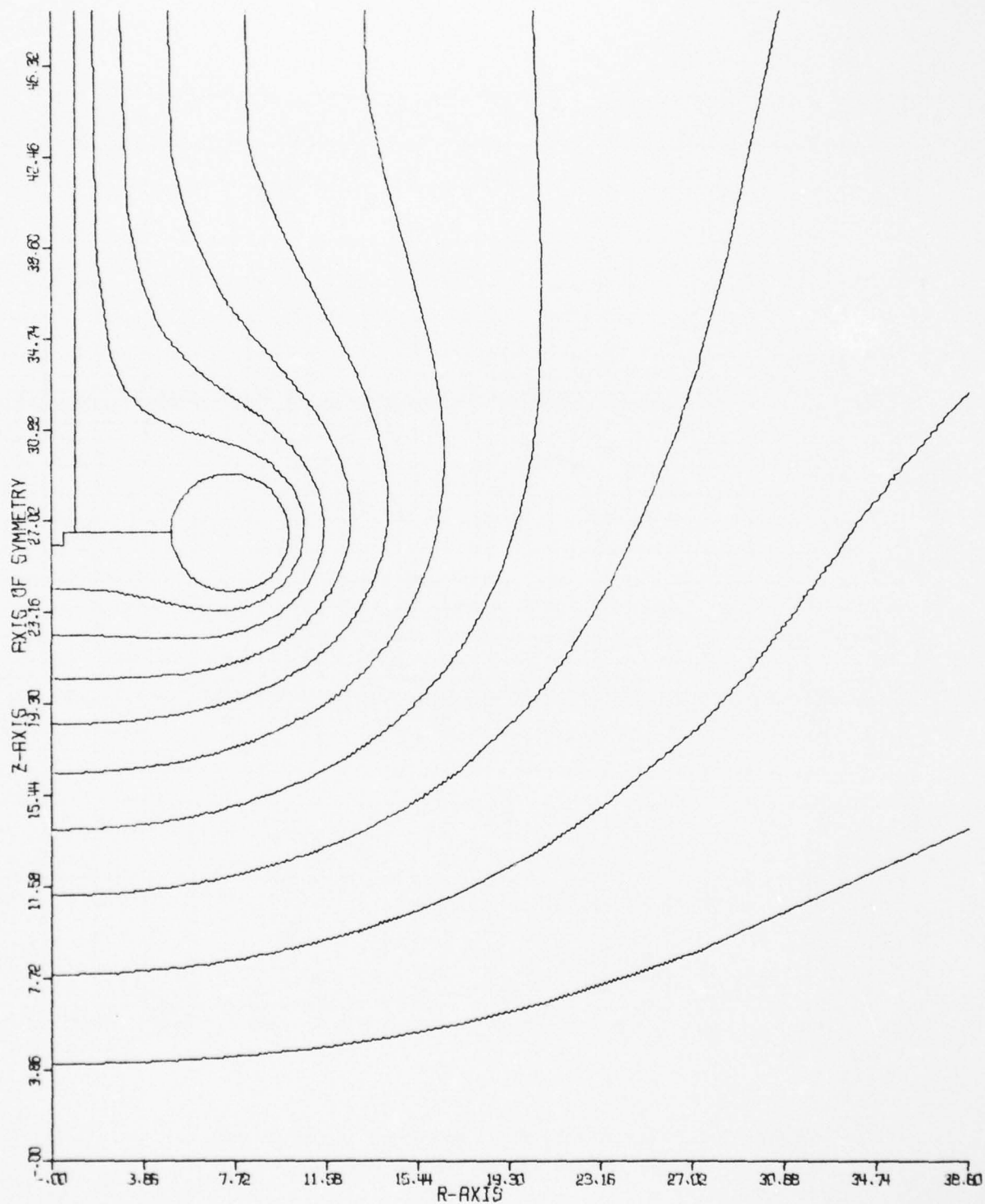


Figure A-3. Plot of equipotential lines in the neighborhood of the corona ring.

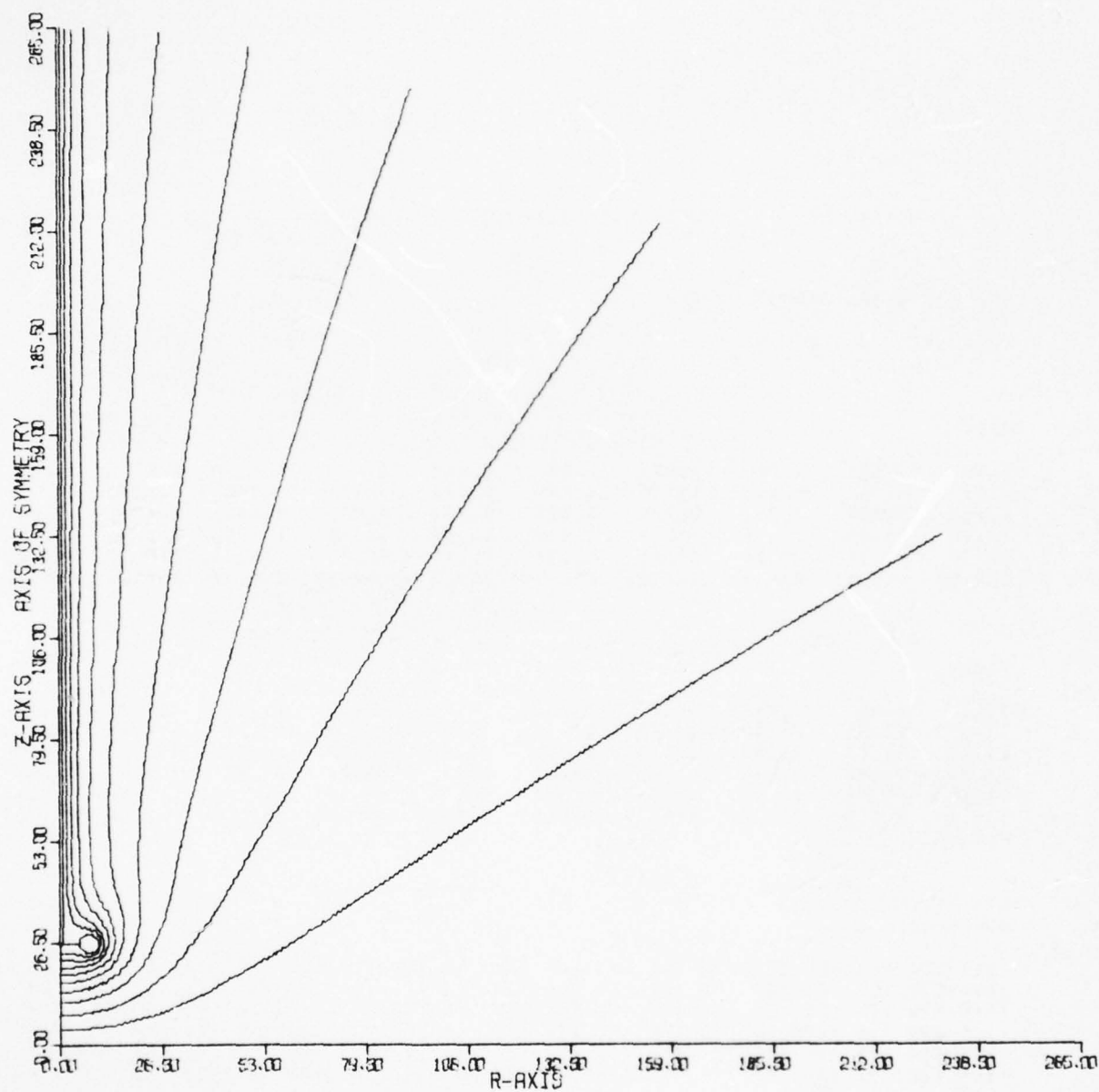


Figure A-4. Plot of equipotential lines for the entire solution region of the corona ring problem.

TABLE A-1. INPUT DATA FOR FVSOLVR

NX, NY, NEPZ, IFSKIP, IFBOY

60 61 2 2 0 0

MX(I)

0.00	0.50	1.00	1.27	1.53	1.84	2.51	3.18	3.80	4.50
4.85	5.00	5.051	5.179	5.429	5.776	6.144	6.520	6.735	7.003
7.500	7.997	8.265	8.480	8.856	9.224	9.571	9.821	9.949	10.00
10.09	10.26	10.49	10.76	11.09	11.64	12.42	13.23	14.08	15.00
17.20	20.70	27.30	38.60	57.00	71.10	90.20	113.00	135.70	160.40
189.50	218.10	236.30	244.50	252.50	258.50	260.50	264.58	264.85	265.00

MY(J)

0.00	4.00	8.00	13.00	15.00	19.00	20.80	21.12	21.87	22.35
22.89	23.21	23.50	23.75	23.90	24.00	24.05	24.12	24.20	24.40
24.69	25.10	25.57	26.00	26.50	27.00	27.43	27.90	28.31	28.60
28.80	28.88	28.95	29.00	29.12	29.38	29.74	30.28	30.89	31.59
32.50	33.20	34.00	35.00	48.63	58.33	80.43	102.20	119.95	150.52
185.24	210.94	227.62	239.70	249.18	255.28	258.80	262.17	263.60	264.50
265.00									

BDR(I,J)

118	120	118	120	118	120	118	120	117	120	116	120	116	120	116	120	116	120	116	120
116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120
116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120
116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120	116	120
116	120	116	120	115	120	114	120	113	120	111	120	109	120	107	120	105	120	103	120
101	120	99	120	97	120	95	120	93	120	91	120	89	120	87	120	85	120	102	120

KO, KN, KK

1 59 2

ESPR(KK)

1.0 0.0

TABLE A-1. (cont.)

KO, KN, KK

60 60 1

ESPR(KK)

0.0

BOZ(I,J)

120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122
120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122
120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122
120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122	120	122
120	122	119	122	117	122	115	122	113	122	111	122	109	122	107	122	105	122	103	122
101	122	99	122	97	122	95	122	93	122	91	122	87	122	11	122	122	122		

KO, KN, KK

1 58 2

ESPZ(KK)

1.0 0.0

KO, KN, KK

59 59 1

ESPZ(KK)

0.0

NPUTS

2

TABLE A-1. (cont.)

POT

0.0

NLINES

1

L1, L2, M1, M2

1 60 1 1

POT

100.

NLINES

21

L1, L2, M1, M2

1	2	24	61
3	3	25	61
4	12	25	25
13	13	24	26
14	14	23	27
15	15	22	28
16	16	21	29
17	17	20	30
18	18	19	31
19	19	18	32
20	20	17	33
21	21	16	34
22	22	17	33
23	23	18	32
24	24	19	31
25	25	20	30
26	26	21	29
27	27	22	28
28	28	23	27
29	29	24	26
30	30	25	25

TABLE A-2. INPUT DATA FOR FVPLT FOR FIGURE A-3

IVU

1

IO, IN, JO, JN, NQUAD, KK, VMIN, SZ

1 44 1 45 1 9 0.0 10.0

NX, NY

(DATA ON TAPE FROM FVSOLVR)

HX(I), HY(J)

(DATA ON TAPE FROM FVSOLVR)

V(I,J)

(DATA ON TAPE FROM FVSOLVR)

NLINES

3

NPTS

2

LX(N), LY(N)

1 1 44 1

TABLE A-2. (Cont.)

NPTS

5

LX(N), LY(N)

1 24 2 24 2 25 3 25 3 45

NPTS

38

LX(N), LY(N)

3	25	12	25	13	26	14	27	15	28	16	29	17	30	18	31	19	32	20	33
21	34	22	33	23	32	24	31	25	30	26	29	27	28	28	27	29	26	30	25
29	24	28	23	27	22	26	21	25	20	24	19	23	18	22	17	21	16	20	17
19	18	18	19	17	20	16	21	15	22	14	23	13	24	12	25				

Appendix B

PROGRAM LISTINGS

The Fortran program listings for FVSOLVR and FVPLLOT which follow are current as of the date of publication of this report. These listings supersede those given in CEL Technical Note N-1502. Revisions are made periodically as deemed necessary; however these changes are usually concerned with minor details. Several versions of FVSOLVR and FVPLLOT exist. For sake of brevity only one version of each is listed here.

The version of FVSOLVR given here is applicable to geometries which have axial symmetry. Other versions apply to two-dimensional cartesian coordinate systems which may or may not have planes of symmetry.

FVSOLVR uses three tapes (or disks) during execution. Tape 33 and tape 34 are used only as "scratchpads" during execution of the algorithm. Upon completion of the program execution tape 10 contains NX, NY, HX(NX), HY(NY), and V(NX,NY), the last array being the nodal solutions.

The version of FVPLLOT given here yields Calcomp plots for 11-in.-wide paper. The plots are drawn such that the smallest dimension of a rectangular plot is scaled to a width of 10 inches. Field solution data from FVSOLVR are read from tape 10 (potential solution data) and tape 12 (flux solution data).

For completeness, the optional data-check program listings are provided. BDYCHK aids in verification of placement of dielectric boundaries. EPSCHK aids in verification of assignment of dielectric constants to the appropriate regions.

```

C      PROGRAM FVSOLVR
C
C      FIELD VALUE SOLVER (AXIAL SYMMETRY)
C
      DIMENSION HX(100),HY(200),HW(99),HM(200),AN(99),
1 EPSR(199,10),EPS7(99,10),ANEPS(99),RHN(99),SYM(200,4),
27EPS1(99),V(100,200),R(200),AK(100,200)
      DIMENSION GH(200),VG(200)
      INTEGER RDR(199,10),RD7(99,10)
      INTEGER BLNK
      COMMON/RANARG/NDPN,NHMLK,R,AK,NN,NN2,V,NX,NY,LX,LV,M,NCKR
      COMMON/HDRARG/RDR,EPSE
      COMMON/VERARG/RD7,EPSE7
      COMMON/SPARG/GH,VG,HM
      DATA BLNK/4H /
      DATA AR/20000*0.0/,SYM/R00*0.0/,R/200*0.0/,V/20000*1.0/
C
C      FORMAT STATEMENTS
C
1 FORMAT(10F8.0)
2 FORMAT(20I4)
3 FORMAT(16F5.0)
4 FORMAT(4F10.3)
5 FORMAT(1H1,4HNY =,T4,4X,4HNY =,T4)
6 FORMAT(1H ,A1,3HMX(,T3,3H) =,F8.3,4(A4,6X,3HMX(,T3,3H) =,F8.3))
7 FORMAT(1H ,A1,3HMY(,T3,3H) =,F8.3,4(A4,6X,3HMY(,T3,3H) =,F8.3))
8 FORMAT(1H1,9X,1H+,14F8.3)
9 FORMAT(10X,1H+,14F8.3)
10 FORMAT(10X,1H+,14(AH-----+---1))
12 FORMAT(1H0,15HINITIAL VALUE =,F8.3)
13 FORMAT(1H0,6HPTS =,T7)
14 FORMAT(1X,7HNUMBLK=,T5)
15 FORMAT(1X,5HCK5=,T5)
16 FORMAT(1X,6HTEST=,T5)
17 FORMAT(13H NONSYMMETRIC)
18 FORMAT(1H1)
19 FORMAT(1H )
20 FORMAT(1H0)
21 FORMAT(1H-)
22 FORMAT (1H ,27HREADY TO BEGIN CALCULATIONS)
23 FORMAT(21A,4F12.3)
24 FORMAT(1H0,T4,36H ERRORS IN SEQUENCE OF HX OR HY DATA)
C
C      INITIALIZATION
C
      READ 2,NX,NY,NRPR,NRZ,IFSKIP,IFRDY
      PRINT 5 ,NX,NY
      PRINT 20
      NYM1=NY-1
      NYM1=NY-1
      NNNY
      NN2=NN+2
      NNPI=NN+1
      REWIND 10
      REWIND 33
      REWIND 34
C
C      READ GRID DATA
C
      READ 1 ,(HX(I),I=1,NX)

```

THIS PAGE IS BEST QUALITY FRAGMENT
FROM COPY FURNISHED TO DDC

```

      K1=0
      NCOL=NX*1/50
      GOTO 120
100  NCOL=NCOL-1
120  INK=NCOL*50
130  K1=K1+1
      K2=K1+INK
      PRINT 6, (RLNK, K, HY(K), K=K1, K2, 50)
      IF (K1.EQ.50.OR.K1.EQ.NX) GOTO 140
      IF (K2.EQ.NX) 100, 130
140  READ 1, (HY(I), I=1, NY)
      PRINT 18
      K1=0
      NCOL=NY*1/50
      GOTO 170
150  NCOL=NCOL-1
170  INK=NCOL*50
180  K1=K1+1
      K2=K1+INK
      PRINT 7, (RLNK, K, HY(K), K=K1, K2, 50)
      IF (K1.EQ.50.OR.K1.EQ.NY) GOTO 200
      IF (K2.EQ.NY) 150, 180
200  READ 2, ((RD0(I, J), J=1, NFPW), I=1, NY*1)
220  READ 2, KN, KK, KK
      READ 3, (FPSR(KN, K), K=1, KK)
      IF (KN.EQ.KN) GOTO 260
      KI=KN+1
      DO 240 I=KI, KN
      DO 240 K=1, KK
240  FPSR(I, K)=FPSR(KN, K)
260  IF (KK.EQ.NFPW) GOTO 300
      KP=KK+1
      DO 280 I=KN, KN
      DO 280 J=KP, NFPW
280  FPSR(I, J)=FPSR(I, KK)
300  IF (KN.LT.NY*1) GOTO 220
      READ 2, ((RDZ(I, J), J=1, NFPZ), I=1, NX*1)
320  READ 2, KN, KK, KK
      READ 3, (FPSZ(KN, K), K=1, KK)
      IF (KN.EQ.KN) GOTO 360
      KI=KN+1
      DO 340 I=KI, KN
      DO 340 K=1, KK
340  FPSZ(I, K)=FPSZ(KN, K)
360  IF (KK.EQ.NFPZ) GOTO 400
      KP=KK+1
      DO 380 I=KN, KN
      DO 380 J=KP, NFPZ
380  FPSZ(I, J)=FPSZ(I, KK)
400  IF (KN.LT.NX*1) GOTO 320
      IF (IFRDY.EQ.0) GOTO 550

```

C
 C
 C
 C

```

      INTERPOLATE FINE GRID BOUNDARY POTENTIALS
      FROM COARSE GRID POTENTIAL SOLUTION

      READ 2, NGX, NGY
      READ 4, (GH(I), I=1, NGX)
      READ 4, (VG(I), I=1, NGX)
      CALL SPONEF (NGX)
      DO 450 I=1, NX

```

THIS PAGE IS BEST QUALITY PHOTOGRAPH
 FROM COPY FURNISHED TO DDC

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

450 V(I,NY)=SPLINE(NGX,HX(I))
    READ 4,(GH(I),I=1,NGY)
    READ 4,(VG(I),I=1,NGY)
    CALL SPCOFF(NGY)
    DO 500 I=1,NYM1
500 V(NX,I)=SPLINE(NGY,HY(I))
C
C    READ BOUNDARY CONDITIONS
C
550 READ 2,NPTS
    PRINT 1A
    DO 600 N=1,NPTS
    READ 1,POT
    PRINT 12,POT
    READ 2,NLINES
    PRINT 2,NLINES
    DO 600 K=1,NLINES
    READ 2,L1,L2,M1,M2
    PRINT 2,L1,L2,M1,M2
    DO 600 I=L1,L2
    DO 600 J=M1,M2
600 V(I,J)=POT
C
C    PARAMETER INITIALIZATION
C
    NUMBLK=0
    NCK=0
    NCK1=1
    NCK4=NN
    NCK5=0
    NCK6=0
    NCNT=0
    NSYM=0
    LY=2
    Z=(HY(1)+HY(2))/2.
    ZPH=(HY(2)+HY(3))/2.
    NIIMX=NX-1
    IF (HX(1).NE.0.0) NIIMX=NX-2
    NPTS=NIIMX*(NY-2)
    PRINT 20
    PRINT 13,NPTS
    NCK7=NPTS+NUMX
    NDIAG=1
    NDP1=NDIAG+1
    NDPN=NDIAG+NIIMX
    PRINT 1A
    DO 650 K=1,NXM1
650 ZEPS1(K)=EPS7(K,1)
C
C    CALCULATE DISTANCES BETWEEN GRID LINES
C
    DO 700 J=1,NYM1
    HN(J)=HX(J+1)-HX(J)
700 IF (HN(J).LE.0.) NCK6=NCK6+1
    DO 720 J=1,NYM1
    HM(J)=HY(J+1)-HY(J)
720 IF (HM(J).LE.0.) NCK6=NCK6+1
    IF (NCK6.EQ.0) GOTO 740
    PRINT 24,NCK6

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

C      CALCULATE AREAS OF HORIZONTAL FACES, SET RHN ARRAY
C
740  R=HX(1)
    AN(1)=0.5*HN(1)*(R+HN(1)/4.)
    RHN(1)=R+HN(1)/2.
    DO 750 J=2,NXM1
    RHN(J)=(HX(J)+HX(J+1))/2.
750  AN(J)=(RHN(J)*2-RHN(J-1)*2)/2.
C
C      BOTTOM EDGE AREA = EPS DATA SET-UP
C
    EPS1=EPSR(1,1)
    R=HX(1)
    LX=1
    J=1
    CALL HORIZ(1,J,EPS1,R,RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
    DO 800 LX=2,NXM1
    J=1
    CALL HORIZ(1,J,EPS1,RHN(LX-1),RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
800  CONTINUE
    PRINT 22
    IF (IFSKIP.NE.0) GOTO 5000
    IF (NCK6.GT.0) GOTO 6000
    PRINT 20
    GOTO 3000
C
C      SHIFT BLOCK UP AND ZERO
C
900  DO 1000 K=1,NN
    KPNN=K+NN
    DO 950 I=1,4
    SYM(K,I)=SYM(KPNN,I)
950  SYM(KPNN,I)=0.0
    R(KPNN)=0.0
    DO 1000 I=1,NN
1000 AR(I,KPNN)=0.0
C
C      CALCULATE BLOCK OF AR, R ARRAY VALUES
C
3000 DO 3600 L=NNP1,NN2
    IF (NCK.EQ.0) GOTO 3010
    LX=LX+1
    R=HX(LX)
    IF (LX.LT.NX) GOTO 3400
    LY=LY+1
    IF (LY.EQ.NY) GOTO 4000
    Z=7PH
    ZPH=HY(LY)+HM(LY)/2.
3010 J=1
    NCK=NCK+1
    EPS1=EPSR(LY,1)
    IF (HX(1).NE.0.0) GOTO 3200
    LX=1
    R=0.0
    IF (V(LX,LY).GE.0.0) GOTO 3500
    SYM(L,1)=0.0
    SYM(L,3)=ANEPS(1)/HM(LY-1)
    CALL HORIZ(LY,J,EPS1,R,RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
    SYM(L,4)=ANEPS(1)/HM(LY)
    CALL VERT(LX,7EPS1(LX),Z,ZPH,RHN(LX),A7EPS,LV,HY(LY))

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

SYM(L,2)=A7EPS/HN(LX)
AR(NDIAG,L)=SYM(L,1)-SYM(L,2)-SYM(L,3)-SYM(L,4)
IF (V(LX,LY=1).LT.0.0) GOTO 3170
R(L)=V(LX,LY=1)*SYM(L,3)
SYM(L,3)=0.0
3170 IF (V(LX+1,LY).LT.0.0) GOTO 3180
R(L)=R(L)-V(LX+1,LY)*SYM(L,2)
SYM(L,2)=0.0
3180 IF (V(LX,LY+1).LT.0.0) GOTO 3190
R(L)=R(L)-V(LX,LY+1)*SYM(L,4)
SYM(L,4)=0.0
3190 AR(NDP1,L)=SYM(L,2)
AR(NDPN,L)=SYM(L,4)
GOTO 3400
3200 LX=2
R=HX(LX)
IF (V(LX,LY).GE.0.0) GOTO 3330
SYM(L,3)=ANEPS(2)/HM(LY=1)
CALL HORIZ(LY,J,FPS1,RHN(LX=1),RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
SYM(L,4)=ANEPS(2)/HM(LY)
CALL VERT(LX=1,7FPS1(LX=1),Z,ZPH,RHN(LX=1),A7EPS,LY,HY(LY))
SYM(L,1)=A7EPS/HN(LX=1)
CALL VERT(LX,7FPS1(LX),Z,ZPH,RHN(LX),A7EPS,LY,HY(LY))
SYM(L,2)=A7EPS/HN(LX)
AR(NDIAG,L)=SYM(L,1)-SYM(L,2)-SYM(L,3)-SYM(L,4)
R(L)=V(L,LY)*SYM(L,1)
SYM(L,1)=0.0
IF (V(LX,LY=1).LT.0.0) GOTO 3270
R(L)=R(L)-V(LX,LY=1)*SYM(L,3)
SYM(L,3)=0.0
3270 IF (V(LX+1,LY).LT.0.0) GOTO 3280
R(L)=R(L)-V(LX+1,LY)*SYM(L,2)
SYM(L,2)=0.0
3280 IF (V(LX,LY+1).LT.0.0) GOTO 3290
R(L)=R(L)-V(LX,LY+1)*SYM(L,4)
SYM(L,4)=0.0
3290 AR(NDP1,L)=SYM(L,2)
AR(NDPN,L)=SYM(L,4)
GOTO 3400
3300 AR(NDIAG,L)=1.
R(L)=V(LX,LY)
CALL HORIZ(LY,J,FPS1,R,RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
CALL VERT(LX,7FPS1(LX),Z,ZPH,RHN(LX),A7EPS,LY,HY(LY))
GOTO 3400
3330 AR(NDIAG,L)=1.
R(L)=V(LX,LY)
CALL HORIZ(LY,J,FPS1,RHN(LX=1),RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
CALL VERT(LX,7FPS1(LX),Z,ZPH,RHN(LX),A7EPS,LY,HY(LY))
GOTO 3400
3400 IF (V(LX,LY).GE.0.0) GOTO 3330
SYM(L,3)=ANEPS(LX)/HM(LY=1)
CALL HORIZ(LY,J,FPS1,RHN(LX=1),RHN(LX),ANEPS(LX),AN(LX),LX,HX(LX))
SYM(L,4)=ANEPS(LX)/HM(LY)
SYM(L,1)=A7EPS/HN(LX=1)
CALL VERT(LX,7FPS1(LX),Z,ZPH,RHN(LX),A7EPS,LY,HY(LY))
SYM(L,2)=A7EPS/HN(LX)
AR(NDIAG,L)=SYM(L,1)-SYM(L,2)-SYM(L,3)-SYM(L,4)
IF (V(LX=1,LY).LT.0.0) GOTO 3460
R(L)=V(LX=1,LY)*SYM(L,1)
SYM(L,1)=0.0

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

3460 IF (V(LX,LY-1).LT.0.0) GOTO 3470
      R(L)=B(L)-V(LX,LY-1)*SYM(L,3)
      SYM(L,3)=0.0
3470 IF (V(LX+1,LY).LT.0.0) GOTO 3480
      R(L)=B(L)-V(LX+1,LY)*SYM(L,2)
      SYM(L,2)=0.0
3480 IF (V(LX,LY+1).LT.0.0) GOTO 3490
      R(L)=B(L)-V(LX,LY+1)*SYM(L,4)
      SYM(L,4)=0.0
3490 AR(NDP1,L)=SYM(L,2)
      AR(NDPN,L)=SYM(L,4)
3600 CONTINUE
4000 NTEST=3
C
C      WRITE BLOCK OF AR, R ARRAY VALUES ONTO TAPE
C
      WRITE(34)(R(N),AR(NDTAG,N),AR(NDP1,N),AR(NDPN,N),N=NNP1,NA2)
      NTEST=4
      NIIMBLK=NIIMBLK+1
      NCNT=NIIMBLK*NN
      IF(NIIMBLK.EQ.1)GOTO 4300
C
C      CHECK MATRIX SYMMETRY
C
4100 DO 4200 L=NCK1,NCK4
      NCK5=NCK5+1
      NCK2=L+1
      NCK3=L+NUMX
      IF (SYM(L,2).NE.SYM(NCK2,1)) GOTO 4150
      IF(NCK5.GT.NCK7)GOTO 4200
      IF (SYM(L,4).EQ.SYM(NCK3,3)) GOTO 4200
4150 PRINT 17
      NSYM=NSYM+1
      PRINT 23,NIIMBLK,NCK5,SYM(NCK2,1),SYM(L,2),SYM(NCK3,3),SYM(L,4)
4200 CONTINUE
      IF(NCK1.EQ.NNP1)GOTO 4500
4300 IF(NCNT.LT.NPTS)GOTO 900
      NCK1=NNP1
      NCK4=NNP1+NPTS-NCK5-1
      GOTO 4100
4500 CONTINUE
      PRINT 15,NCK5
      NCK8=NPTS-(NIIMBLK-1)*NN
      PRINT 2,NCK8
      IF (NSYM.GT.0) GOTO 5000
C
C      ZERO AR AND R ARRAYS
C
      DO 4800 J=1,NN2
      R(J)=0.0
      DO 4800 I=1,NN
4800 AR(L,J)=0.0
C
C      CALL Banded Matrix Solver
C
      M=1
      IF (HX(1).NE.0.0) M=2
      CALL BANSOL(AR,NDPN,NN2)
      NTEST=5
      PRINT 2,LX,LY

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

C
C
C      WRITE SOLUTION ONTO TAPE
C
C      WRITE(10) NX,NY
C      CALL WRT10(HX,NX)
C      CALL WRT10(HY,NY)
C      DO 4900 J=1,NY
4900  CALL WRT10(V(1,J),NX)
C      END FILE10
C
C      PRINT SOLUTION
C
5000  M2=0
C      M1=50
5100  M2=M2+50
C      IF (M2.LE.NY) GOTO 5200
C      M1=NY+50-M2
C      M2=NY
5200  N2=0
5300  N1=N2+1
C      N2=N2+14
C      IF (N2.GT.NX) N2=NX
C      NPRT=N2-N1+1
C      PRINT 8,(HX(K),K=N1,N2)
C      PRINT 10
C      DO 5400 J=1,M1
C      I=M2+1-J
5400  CALL PRTY(HY(I),V(N1,I),NPRT)
C      PRINT 10
C      PRINT 9,(HY(K),K=N1,N2)
C      IF (N2.LT.NX) GOTO 5300
C      IF (M2.LT.NY) GOTO 5100
6000  CONTINUE
C      STOP
C      END
C
C      SUBROUTINE RANSOL(A,N1,N2)
C      SOLVES Banded, SYMMETRIC MATRIX
C
C      DIMENSION R(200),AR(100,200),V(100,200)
C      DIMENSION A(N1,N2)
C      COMMON /RANAR/ MM,NUMR(K,R,AK,NN,NN2,V,NX,NY,LX,LY,LHR,NCKB
C      NL=NN+1
C      NH=NN+NN
C      LY=NY
C      LY=NY-1
C      REWIND 34
C      NB=0
C      GO TO 150
C
C
C      REDUCE EQUATIONS BY BLOCKS
C
C
C      1. SHIFT BLOCK OF EQUATIONS
C
100  NR=NR+1
C      DO 125 N=1,NN
C      NM=NN+N
C      R(N)=R(NM)
C      R(NM)=0.0

```

THIS PAGE IS BEST QUALITY ~~FRAGMENTS~~
FROM COPY FURNISHED TO DDC

```

      DO 125 M=1,MM
      A(M,N)=A(M,MM)
125  A(M,MM)=0.0
C
C      2. READ NEXT BLOCK OF EQUATIONS INTO CORE
C
      IF (NUMBLK=NR) 150,200,150
150  READ (34) (R(N),A(1,N),A(2,N),A(MM,N),N=NL,NH)
      IF (NR) 200,100,200
C
C      3. REDUCE BLOCK OF EQUATIONS
C
      DO 300 N=1,NN
      IF (A(1,N)) 225,300,225
225  R(N)=R(N)/A(1,N)
      DO 275 L=2,MM
      IF (A(L,N)) 230,275,230
230  C=A(1,N)/A(L,N)
      T=N+L-1
      J=0
      DO 250 K=L,MM
      J=J+1
250  A(J,T)=A(J,T)-C*A(K,N)
      R(T)=R(T)-A(L,N)*R(N)
      A(L,N)=C
275  CONTINUE
300  CONTINUE
C
C      4. WRITE BLOCK OF REDUCED EQUATIONS ON TAPE 2
C
      IF (NUMBLK=NR) 375,400,375
375  CALL RDWRPT33(R,A(1,1),NN,MM,1)
      GO TO 100
C
C      BACK-SUBSTITUTION
C
      DO 450 M=1,NN
      N=NN+1-M
      DO 425 K=2,MM
      L=N+K-1
425  R(N)=R(N)-A(K,N)*R(L)
      N=NN+NN
      R(NM)=R(N)
      IF (NR.EQ.NUMBLK.AND.N.GT.NCKB) GOTO 450
      LX=LX+1
      IF (LX.GE.LHR) GOTO 445
      LY=LY+1
      LX=NX-1
445  V(LX,LY)=R(N)
450  CONTINUE
      NR=NR+1
      IF (NR) 475,500,475
475  CONTINUE
      CALL RDWRPT33(R,A(1,1),NN,MM,2)
      GO TO 400
500  CONTINUE
C
C      RETURN
C
      END

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

C      SUBROUTINE RDWRIT33(B,A,N,M,INDX)
        DIMENSION R(N),A(M,N)
        DIMENSION MIDX(400)
        DATA NW/0/
        IF(NW.NE.0) GO TO 5
        CALL OPENMS(33,MIDX,400,0)
        KR=N
        KA=N*M
5      IF(INDX.NE.1) GO TO 10
C      WRITE
        NW=NW+1
        CALL WRITMS(33,B,KR,NW)
        NW=NW+1
        CALL WRITMS(33,A,KA,NW)
        GO TO 20
C      READ
10     CALL READMS(33,A,KA,NW)
        NW=NW+1
        CALL READMS(33,B,KR,NW)
        NW=NW+1
C      DONE
20     CONTINUE
        RETURN
        END

C      SUBROUTINE WRT10(A,N)
        DIMENSION A(N)
        WRITE(10) A
        RETURN
        END

C      SUBROUTINE PRTY(A,B,N)
        DIMENSION R(N)
        PRINT 11,A,B
11     FORMAT(1X,F8.3,2H +,14F8.3)
        RETURN
        END

C      SUBROUTINE HORIZ(LY,J,EPS1,RA,RB,ANEPS,AREA,LX,R)
C      CALCULATES DIELECTRIC-AREA PRODUCT
C      OF HORIZONTAL FACE OF VOLUME ELEMENT
C
        DIMENSION EPS(100,10)
        INTEGER RDR(100,10)
        COMMON/HORRARG/RDR,EPS
        IF (RDR(LY,J)-2*LX) 1,2,3
1      J=J+1
        EPS1=EPS(LY,J)
        IF (RDR(LY,J)-2*LX) 1,2,3
2      J=J+1
        EPS2=EPS(LY,J)
        ANEPS=0.5*((EPS1-EPS2)*R**2+EPS2*R**2-EPS1*R**2)
        EPS1=EPS2
        RETURN
3      ANEPS=EPS1*AREA
        RETURN
        END

C      SUBROUTINE VERT(LX,EPS1,Z,ZPH,R,AZEPS,LY,Y)

```

```

C   CALCULATES DIELECTRIC-AREA PRODUCT
C   OF VERTICAL FACE OF VOLUME ELEMENT
C
      DIMENSION EPS(99,10)
      INTEGER RD7(99,10)
      COMMON/VERARG/RD7, EPS
      K=1
      IF (RD7(LX,K)-2*LY) 1,2,3
1     K=K+1
      EPS1=EPS(LX,K)
      IF (RD7(LX,K)-2*LY) 1,2,3
2     K=K+1
      EPS2=EPS(LX,K)
      AZEPS=R*(EPS1*(Y-Z)+EPS2*(7PH-Y))
      EPS1=EPS2
      RETURN
3     AZEPS=EPS1*R*(7PH-7)
      RETURN
      END

C   SUBROUTINE SPCDEF (N)
C   CALCULATES COEFFICIENTS OF CURIC SPLINE USED IN INTERPOLATION
C
      DIMENSION XN(200), FN(200), S(200), RHO(200), TAU(200)
      COMMON/SPARG/XN, FN, S
      NM1=N-1
      NM2=N-2
      RHO(2)=0.0
      TAU(2)=0.0
      DO 1 I=2, NM1
      TM1=I-1
      TP1=I+1
      HM1=XN(T)-XN(TM1)
      HI=XN(TP1)-XN(T)
      TEMP=(HM1/HT1*(RHO(T)+2.))+2.
      RHO(I+1)=1./TEMP
      D=6.*((FN(TP1)-FN(T))/HT1-(FN(T)-FN(TM1))/HM1)/HT1
1     TAU(I+1)=(D-HM1*TAU(T))/HT1/TEMP
      S(1)=0.
      S(N)=0.
      DO 2 I=1, NM2
      TR=N-I
2     S(TR)=RHO(TR+1)*S(TR+1)+TAU(TR+1)
      RETURN
      END

C   FUNCTION SPLINE (N,X)
C   EVALUATES SPLINE FUNCTION
C
      DIMENSION XN(200), FN(200), S(200)
      COMMON/SPARG/XN, FN, S
      IF (X.GE.XN(1)) GOTO 1
      H1=XN(2)-XN(1)
      SPLINE=FN(1)+(X-XN(1))*((FN(2)-FN(1))/H1-H1*S(2)/6.)
      RETURN
1     IF (X.LE.XN(N)) GOTO 2
      HN=N-1
      HN1=XN(N)-XN(HN1)
      SPLINE=FN(N)+(X-XN(N))*((FN(N)-FN(HN1))/HN1+HN1*S(N)/6.)
      RETURN

```

```

2 DO 3 I=2,N
  IF (X.IE.XN(I)) GOTO 4
3 CONTINUE
4 I=I+1
  LP=I+1
  A=XN(LP)-Y
  R=X-XN(I)
  HL=XN(LP)-XN(I)
  SPLINE=A*S(I)*(A**2/HL-HL)/6.+R*S(I+1)*(R**2/HL-HL)/6.
/   +(A*R*(I)+R*R*(LP))/HL
  RETURN
END

```

```

      PROGRAM EVPL0T
C
C      CONTOUR LINE PLOTTER
C
      DIMENSION V(100,200),HX(100),HY(200),IPUF(60),GX(100),GY(200),
      /U(100,200),LX(150),LY(150)
      COMMON/DRAWARC/X(600),Y(600),XMAX,YMAX,NQUAD,FIRSTX,DELTAX,FIRSTY,
      /DELTAY
      1 FORMAT(AF10.3)
      2 FORMAT (20I4)
      3 FORMAT(6I4,2F5.0)
      4 FORMAT(I4)
      5 FORMAT (5H1NX =,I4,6X,4HNY =,I4)
      6 FORMAT (1H0,I13,I14,I24,I14)
      7 FORMAT (7H0X FROM,F10.3,4H TO,F10.3,8X,6HY FROM,F10.3,4H TO,
      IF10.3)
C
C      READ POTENTIAL GRID DATA
C
      READ 4,IVU
      READ 3,IO,IN,JO,JN,NQUAD,*,K,VMIN,SZ
      UPOT=SZ
      READ (10) NX,NY
      CALL RD10(HX,NX)
      CALL RD10(HY,NY)
      DO 80 J=1,NY
      80 CALL RD10(V(1,J),NX)
      TSCAN=1
      IF (IVU.EQ.1) GOTO 150
C
C      READ FLUX GRID DATA
C
      READ (12) MX,MY
      CALL RD10(GX,MX)
      CALL RD10(GY,MY)
      DO 90 J=1,MY
      90 CALL RD10(U(1,J),MX)
      NH=1
      NH=2
      KPA=1
C
C      SHIFT DATA FOR REGION TO BE PLOTTED
C
      IF (IO.EQ.1) GOTO 120
      DO 100 I=IO,IN
      GX(I+1-JO)=GX(I)
      DO 100 J=1,MY
      100 U(I+1-JO,J)=U(I,J)
      120 MX=IN+1-IO
      IF (JO.EQ.1) GOTO 150
      DO 140 J=JO,JN
      GY(J+1-JO)=GY(J)
      DO 140 I=1,MX
      140 U(I,J+1-JO)=U(I,J)
      150 IF (IO.EQ.1) GOTO 170
      DO 160 I=IO,IN
      HX(I+1-JO)=HX(I)
      DO 160 J=1,NY
      160 V(I+1-JO,J)=V(I,J)
      170 NX=IN+1-IO

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

      IF (J0.EQ.1) GOTO 200
      DO 180 J=J0,J1
      HY(J+1-J0)=HY(J)
      DO 180 I=1,NX
180   V(I,J+1-J0)=V(I,J)
200   NY=JN+1-J0
      PRINT 5,NX,NY
      PRINT 6,I0,YN,J0,JN
      PRINT 7,HX(1),HX(NX),HY(1),HY(NY)
      YMAX=HX(NX)-HX(1)
      YMAX=HY(NY)-HY(1)
      NYM1=NY-1
      NYM1=NY-1
C
C   INITIALIZE PLOT AND DRAW AXES
C
      CALL PLOTS (11,0,99)
      CALL PLOT (0.,-14.,-3)
      CALL PLOT(0.,0.5,-3)
      FIRSTX=HX(1)
      FIRSTY=HY(1)
      IF (XMAX.GT.YMAX) GOTO 220
      IF (NQIAD.EQ.2.AND.(2.*XMAX).GT.YMAX) GOTO 220
      DELTAX=XMAX/10.
      DELTAY=DELTAY
      GOTO 240
220   DELTAX=YMAX/10.
      DELTAX=DELTAY
      IF (NQIAD.EQ.2) GOTO 290
240   IF (NQIAD.GT.1) GOTO 280
      IF (XMAX.GT.YMAX) GOTO 260
      YAX=(YMAX/YMAX)*10.
      CALL AXIS(0.,10.,27H7-AXIS      AXIS OF SYMMETRY,+27,YAX,0.,FIRSTY,
1      DELTAY,0)
      CALL AXIS(0.,10.,4HR-AXIS,-6,10.,270.,FIRSTX,DELTAY,0)
      GOTO 290
260   YAX=(YMAX/YMAX)*10.
      CALL AXIS(0.,0.,27H2-AXIS      AXIS OF SYMMETRY,+27,10.,90.,FIRSTY,
1      DELTAY,0)
      CALL AXIS(0.,0.,4HR-AXIS,-6,YAX,0.,FIRSTX,DELTAY,0)
      GOTO 290
280   DELTAX=2.*DELTAY
      DELTAY=2.*DELTAY
290   IF (TVU.EQ.2) GOTO 305
C
C   DRAW EQUIPOTENTIAL LINES
C
      DO 300 K=1,KK
      ZEVMIN+K*SZ
300   CALL DRAW (7,NX,NY,NYM1,NYM1,HX,HY,V,ISCAN)
      IF (TVU.EQ.3) 305,500
C
C   CALCULATE AND PLOT FLUX LINE DISTRIBUTION
C
305   NYM1=NY-1
      NYM1=NY-1
      T=1
      PA=V(1,1)
310   T=T+1
      PB=V(1,1)

```

```

IF (UPDT.GE.PA.AND.UPDT.LE.PR) GOTO 320
PA=PR
GOTO 310
320 IM=I-1
R=HY(IM)+(UPDT-PA)*(HY(I)-HY(IM))/(PR-PA)
H=HY(NH)-HY(MH)
TSCAN=2
C=0.
RA=HY(KRA)
330 KRR=KRA+1
IF (KRR.GT.MX) GOTO 500
QB=HY(KRR)
FA=(V(KRA,NH)-V(KRA,MH))/H
FR=(V(KRR,NH)-V(KRR,MH))/H
A2=((FR-FA)/(RR-RA))/2.
A1=EA-RA*2.*A2
CINC=A2*(RR*2-RA*2)+A1*(RH-RA)
IF (R.LT.RR) CINC=A2*(R*2-RA*2)+A1*(R-RA)
C=C+CINC
IF (R.LE.PR) GOTO 340
KRA=KRR
RA=RR
GOTO 330
340 Z=U(KRA,MH)+(R-RA)*(U(KRR,MH)-U(KRA,MH))/(RR-RA)
CALL DRAW (Z,NY,NV,NXM1,NYM1,GX,GY,U,TSCAN)
350 RX=R
CSUM=0.
360 A2=((FR-FA)/(RR-RA))/2.
A1=EA-RA*2.*A2
CINC=A2*(RR*2-RX*2)+A1*(RH-RX)
IF (C-CSUM-CINC) 380,390,370
370 CSUM=CSUM+CINC
KRA=KRR
KRR=KRA+1
IF (KRR.GT.MX) GOTO 500
RA=RR
RX=RA
QB=HY(KRR)
FA=(V(KRA,NH)-V(KRA,MH))/H
FR=(V(KRR,NH)-V(KRR,MH))/H
GOTO 360
380 A0=CSUM-(C+A2+RX*2+A1+RX)
CALL RESLV(R,RX,RR,A2,A1,A0)
GOTO 400
390 R=RR
400 Z=U(KRA,MH)+(R-RA)*(U(KRR,MH)-U(KRA,MH))/(RR-RA)
CALL DRAW (Z,NY,NV,NXM1,NYM1,GX,GY,U,TSCAN)
IF (TSCAN.EQ.0) GOTO 500
IF (R.NE.PR) GOTO 350
KRA=KRR
KRR=KRA+1
IF (KRR.GT.MX) GOTO 500
QB=HY(KRR)
RA=R
GOTO 350

```

C
C
C DRAW ELECTRODE AND DIELECTRIC CONFIGURATION

500 READ 3,NLINES
DO 700 L=1,NLINES

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

READ 3 ,NPTS
READ 2 ,(LX(N),LY(N),N=1,NPTS)
IF (YMAX.GT.YMAX) GOTO 600
IF (NQUAD.EQ.2.AND.(2.*YMAX).GT.YMAX) GOTO 600
DO 540 N=1,NPTS
LXNNN=LX(N)+1-ID
LYNNN=LY(N)+1-JD
V(N)=HY(LYNNN)
GOTO (510,530,520,520),NQUAD
510 X(N)=XMAX-HX(LXNNN)
GOTO 540
520 Y(N)=YMAX-HY(LYNNN)
530 X(N)=XMAX+HX(LXNNN)
540 CONTINUE
X(NPTS+1)=FIRSTX
X(NPTS+2)=DELTAX
Y(NPTS+1)=FIRSTY
Y(NPTS+2)=DELTAY
CALL LINE (Y,X,NPTS,1,0,0)
GOTO (700,580,550,550),NQUAD
550 DO 560 N=1,NPTS
560 X(N)=2.*YMAX-X(N)
CALL LINE (Y,X,NPTS,1,0,0)
DO 570 N=1,NPTS
570 Y(N)=2.*YMAX-Y(N)
CALL LINE (Y,X,NPTS,1,0,0)
580 DO 590 N=1,NPTS
590 X(N)=2.*YMAX-X(N)
CALL LINE (Y,X,NPTS,1,0,0)
GOTO 700
600 DO 640 N=1,NPTS
LXNNN=LX(N)+1-ID
LYNNN=LY(N)+1-JD
V(N)=HY(LYNNN)
GOTO (610,630,620,620),NQUAD
610 X(N)=HX(LXNNN)
GOTO 640
620 Y(N)=YMAX-HY(LYNNN)
630 X(N)=XMAX+HX(LXNNN)
640 CONTINUE
X(NPTS+1)=FIRSTX
X(NPTS+2)=DELTAX
Y(NPTS+1)=FIRSTY
Y(NPTS+2)=DELTAY
CALL LINE (X,Y,NPTS,1,0,0)
GOTO (700,680,650,650),NQUAD
650 DO 660 N=1,NPTS
660 X(N)=2.*YMAX-X(N)
CALL LINE (X,Y,NPTS,1,0,0)
DO 670 N=1,NPTS
670 Y(N)=2.*YMAX-Y(N)
CALL LINE (X,Y,NPTS,1,0,0)
680 DO 690 N=1,NPTS
690 X(N)=2.*YMAX-X(N)
CALL LINE (X,Y,NPTS,1,0,0)
700 CONTINUE
CALL PLOT(12,0,0,0,40)
STOP
END

```

```
SUBROUTINE RD10(A,N)
  DIMENSION A(N)
  READ (10) A
  RETURN
END
```

C

```
SUBROUTINE DRAW (Z,IL,JL,II,JJ,XV,YV,V,ISCAN)
  SCANS GRID AND PLOTS LINES OF EQUAL VALUE
```

C

C

```
  DIMENSION XV(100),YV(200),V(100,200)
  COMMON/DRAW/ARG/X(600),Y(600),XMAX,YMAX,NQUAD,FIRSTX,DELTAX,FIRSTY,
  /DELTAY
  NT=0
  TX=0
  T=7
  TA=IL
  DO 3 J=1,JJ
    JA=J
    A=V(TA,JA)
    D=V(TA,JA+1)
    IF (A.EQ.D) GO TO 3
    IF (A.LE.0..OR.D.LE.0.) GOTO 5
    IF (A.EQ.T) 1,2
  1 T=A+(D-A)*1.E-6
  2 CALL DA(D,A,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
    IF (NT.NE.0) GO TO 30
  3 CONTINUE
    JA=JL
    DO 6 I=1,II
      IA=IL+I
      R=V(IA+1,JA)
      A=V(IA,JA)
      IF (R.EQ.A) GO TO 6
      IF (A.LE.0..OR.R.LE.0.) GOTO 6
      IF (R.EQ.T) 4,5
  4 T=R+(A-R)*1.E-6
  5 CALL AR(A,R,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (NT.NE.0) GO TO 30
  6 CONTINUE
      TA=0
      DO 10 J=1,JJ
        JA=JL+J
        C=V(1,JA+1)
        R=V(1,JA)
        IF (C.EQ.R) GO TO 10
        IF (JA.EQ.1) GOTO 7
        IF (C.LE.0..OR.R.LE.0.) GOTO 10
  7 IF (C.LE.0..OR.R.LT.0.) GOTO 10
        IF (C.EQ.T) 8,9
  8 T=C+(R-C)*1.E-6
  9 CALL BC(R,C,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
        IF (NT.NE.0) GO TO 30
  10 CONTINUE
      JA=0
      DO 14 I=1,II
        IA=I
        D=V(TA,1)
        C=V(TA+1,1)
        IF (D.EQ.C) GO TO 14
        IF (TA.EQ.1.AND.ISCAN.EQ.2) GOTO 11
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

      IF (C.LE.0..OR.D.LE.0.) GOTO 14
11  IF (C.LE.0..OR.D.LT.0.) GOTO 14
      IF (D.EQ.T) 12,13
12  T=D+(C=D)*1.F=6
13  CALL CD(C,D,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (NT.NE.0) GO TO 30
14  CONTINUE
      IF (TSCAN.NE.1) GOTO 22
      DO 21 I=1,II
      DO 21 J=1,JJ
      TA=I
      JA=J
      C=V(I+1,J+1)
      R=V(I+1,J)
      IF (C.EQ.8) GO TO 18
      IF (J.EQ.1) GOTO 15
      IF (C.LE.0..OR.R.LE.0.) GOTO 18
15  IF (C.LE.0..OR.R.LT.0.) GOTO 18
      IF (C.EQ.T) 16,17
16  T=C+(R=C)*1.F=6
17  CALL RC(R,C,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      IF (NT.NE.0) GO TO 30
18  R=V(I,J+1)
      IF (C.EQ.D) GO TO 21
      IF (C.LE.0..OR.D.LE.0.) GOTO 21
      IF (C.EQ.T) 19,20
19  T=C+(D=C)*1.F=6
20  CALL CD(C,D,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (NT.NE.0) GO TO 30
21  CONTINUE
      TSCAN=0
      GO TO 69
22  DO 29 J=1,JJ
      DO 29 I=1,II
      TA=I
      JA=J
      C=V(I+1,J+1)
      R=V(I+1,J)
      IF (C.EQ.8) GO TO 25
      IF (C.LE.0..OR.R.LE.0.) GOTO 25
      IF (C.EQ.T) 23,24
23  T=C+(R=C)*1.F=6
24  CALL RC(R,C,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      IF (NT.NE.0) GO TO 30
25  R=V(I,J+1)
      IF (C.EQ.D) GO TO 29
      IF (T.EQ.1) GOTO 26
      IF (C.LE.0..OR.D.LE.0.) GOTO 29
26  IF (C.LE.0..OR.D.LT.0.) GOTO 29
      IF (C.EQ.T) 27,28
27  T=C+(D=C)*1.F=6
28  CALL CD(C,D,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (NT.NE.0) GO TO 30
29  CONTINUE
      TSCAN=0
      GO TO 69
30  IST=IA
      JST=JA
31  IF (JA.EQ.0..OR.IA.EQ.II..OR.JA.EQ.0..OR.JA.EQ.JJ) GO TO 50
      NIX=IX

```

```

      TF (T.EQ.V(TA,IA)) V(TA,JA)=V(TA,JA)*1.0001
      TF (T.EQ.V(TA+1,JA)) V(TA+1,JA)=V(TA+1,JA)*1.0001
      TF (T.EQ.V(TA+1,JA+1)) V(TA+1,JA+1)=V(TA+1,JA+1)*1.0001
      TF (T.EQ.V(TA,JA+1)) V(TA,JA+1)=V(TA,JA+1)*1.0001
      A=V(TA,JA)
      R=V(TA+1,JA)
      C=V(TA+1,JA+1)
      D=V(TA,JA+1)
      IF (TA.EQ.1.AND.ISCAN.EQ.2) GOTO 32
      IF (JA.EQ.1) GOTO 33
      IF (A.LE.0..OR.R.LE.0..OR.C.LE.0..OR.D.LE.0.) GOTO 40
32  IF (A.LT.0..OR.R.LE.0..OR.C.LE.0..OR.D.LT.0.) GOTO 50
      GOTO 34
33  IF (A.LT.0..OR.R.LT.0..OR.C.LE.0..OR.D.LE.0.) GOTO 50
34  IF (NT.EQ.1) GO TO 35
      IF (NT.EQ.2) GO TO 36
      IF (NT.EQ.3) GO TO 37
      IF (NT.EQ.4) GO TO 38
      GO TO 69
35  CALL DA(D,A,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      IF (IX.GT.NIX) GOTO 39
      CALL RC(R,C,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      IF (IX.GT.NIX) GOTO 39
      CALL AR(A,R,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      GO TO 39
36  CALL AR(A,R,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (IX.GT.NIX) GOTO 39
      CALL CD(C,D,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (IX.GT.NIX) GOTO 39
      CALL RC(R,C,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      GO TO 39
37  CALL RC(R,C,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      IF (IX.GT.NIX) GOTO 39
      CALL DA(D,A,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
      IF (IX.GT.NIX) GOTO 39
      CALL CD(C,D,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      GO TO 39
38  CALL CD(C,D,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (IX.GT.NIX) GOTO 39
      CALL AR(A,R,T,XV,YV,X,Y,TA,JA,IX,NT,II,V)
      IF (IX.GT.NIX) GOTO 39
      CALL DA(D,A,T,XV,YV,X,Y,TA,JA,IX,NT,JJ,V)
39  IF (TA.EQ.1ST.AND.JA.EQ.JST) 50,31
40  IF (X(IX).EQ.X(IX-1)) GOTO 50
      Q=(Y(IX)-Y(IX-1))/(X(IX)-X(IX-1))
      SR=Y(IX)-S+X(IX)
41  SL=(YV(JA+1)-YV(JA))/(XV(IA)-XV(IA+1))
      SR=(YV(JA+1)-YV(JA))/(XV(IA+1)-XV(IA))
      ORTHL=ABS(SL+1./S)
      ORTHR=ABS(SR+1./S)
      IF (ORTHR.LT.ORTH) GOTO 42
      SRD=YV(JA)-SL*XV(IA+1)
      XR=(SRD-SR)/(S-SL)
      GOTO 43
42  SRD=YV(JA)-SR*XV(IA)
      XR=(SRD-SR)/(S-SR)
43  IX=IX+1
      X(IX)=XR
      Y(IX)=S*X(IX)+SH
50  CALL XPLOTY(X,Y,Z,IX)

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

IF (YMAX,GT,YMAX) GOTO 60
IF (NQUAD,EQ,2,AND,(2.*XMAX),GT,YMAX) GOTO 60
DO 54 I=1,IX
GOTO (51,53,52,52),NQUAD
51 X(I)=XMAX-X(I)
GOTO 54
52 Y(I)=YMAX-Y(I)
53 X(I)=XMAX+X(I)
54 CONTINUE
X(IX+1)=FIRSTX
X(IX+2)=DELTAY
Y(IX+1)=FIRSTY
Y(IX+2)=DELTAY
CALL LINE(Y,X,IX,1,0,0)
GOTO (69,58,55,55),NQUAD
55 DO 56 I=1,IX
56 X(I)=2.*XMAX-X(I)
CALL LINE(Y,X,IX,1,0,0)
DO 57 I=1,IX
57 Y(I)=2.*YMAX-Y(I)
CALL LINE(Y,X,IX,1,0,0)
58 DO 59 I=1,IX
59 X(I)=2.*XMAX-X(I)
CALL LINE(Y,X,IX,1,0,0)
GOTO 69
60 GOTO (63,61,61,61),NQUAD
61 DO 62 I=1,IX
Y(I)=XMAX+X(I)
62 IF (NQUAD,EQ,4) Y(I)=2.*YMAX-Y(I)
63 X(IX+1)=FIRSTX
X(IX+2)=DELTAY
Y(IX+1)=FIRSTY
Y(IX+2)=DELTAY
CALL LINE(Y,X,IX,1,0,0)
GOTO (69,67,64,64),NQUAD
64 DO 65 I=1,IX
65 X(I)=2.*XMAX-X(I)
CALL LINE(Y,X,IX,1,0,0)
DO 66 I=1,IX
66 Y(I)=2.*YMAX-Y(I)
CALL LINE(Y,X,IX,1,0,0)
67 DO 68 I=1,IX
68 X(I)=2.*XMAX-X(I)
CALL LINE(Y,X,IX,1,0,0)
69 CONTINUE
RETURN
END

C
C SUBROUTINE RSOLV (R,RA,RR,A2,A1,A0)
C CALCULATES ROOTS OF QUADRATIC EQUATION
C
IF (A2,EQ,0.) GOTO 3
R=(-A1 + SQRT(A1**2 - 4.*A2*A0)) / (2.*A2)
PRINT 1 ,R,RA,RR,A2,A1,A0
1 FORMAT (4H R =,6E16.8)
IF (R,GE,RA,AND,R,LE,RR) GOTO 2
R=(-A1 - SQRT(A1**2 - 4.*A2*A0)) / (2.*A2)
PRINT 1 ,R,RA,RR,A2,A1,A0
2 CONTINUE
RETURN

```

```

3  QZ=A0/A1
   PRINT 1 ,R,RA,RR,A2,A1,A0
   RETURN
   END

C
   SUBROUTINE XPLDXY(X,Y,Z,IX)
C   PRINTS COORDINATES OF EQUAL VALUE LINES
C
   DIMENSION X(1),Y(1)
   PRINT 3
   PRINT 4, Z
   PRINT 2, IX
   PRINT 5
   PRINT 1, (X(K),Y(K),K=1,IX)
1  FORMAT(1H ,2F15.7,3(2X,2F15.7))
2  FORMAT(1H ,16,7H POINTS)
3  FORMAT(1H )
4  FORMAT(9H VALUE = ,F8.3)
5  FORMAT(1H ,16HCOORDINATE PAIRS)
   RETURN
   END

C
   SUBROUTINE AR(A,R,T,XV,YV,X,Y,IA,JA,IX,NT,II,V)
C   SEARCHES BOTTOM SIDE OF RECTANGLE FOR GIVEN VALUE
C
   DIMENSION XV(1),YV(1),X(1),Y(1),V(100,200)
   IF (A.GT.T.AND.B.LT.T.OR.A.LT.T.AND.B.GT.T) 1,7
1  P1=XV(IA)
   Q1=XV(IA+1)
   XX=(P1+Q1)/2.
   T1=A
   T2=B
   IF (IA.EQ.1) 2,3
2  P1=XV(IA+2)
   T3=YV(IA+2,JA)
   CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
   GO TO 6
3  IF (IA.EQ.II) 4,5
4  P1=XV(IA-1)
   T3=YV(IA-1,JA)
   CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
   GO TO 6
5  IF (V(IA-1,JA).LE.0.) GOTO 2
   IF (V(IA+2,JA).LE.0.) GOTO 4
   P1=XV(IA-1)
   S1=XV(IA+2)
   T3=YV(IA-1,JA)
   T4=YV(IA+2,JA)
   CALL SOL44(XX,T,P1,Q1,P1,S1,T1,T2,T3,T4)
6  NT=1
   IX=IX+1
   X(IX)=XX
   Y(IX)=YV(JA)
   JA=JA-1
7  RETURN
   END

C
   SUBROUTINE RC(R,C,T,XV,YV,X,Y,IA,JA,IX,NT,JJ,V)
C   SEARCHES R.H. SIDE OF RECTANGLE FOR GIVEN VALUE
C

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

      DIMENSION XV(1),YV(1),X(1),Y(1),V(100,200)
      IF (R.GT.T.AND.C.LT.T.OR.R.LT.T.AND.C.GT.T) 1,7
1    P1=YV(JA)
      Q1=YV(JA+1)
      XX=(P1+Q1)/2.
      T1=R
      T2=C
      IF (JA.EQ.1) 2,3
2    Q1=YV(JA+2)
      T3=V(JA+1,JA+2)
      CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
      GO TO 6
3    IF (JA.EQ.IJ) 4,5
4    R1=YV(JA-1)
      T3=V(JA+1,JA-1)
      CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
      GO TO 6
5    IF (V(JA+1,JA-1).LE.0.) GOTO 2
      IF (V(JA+1,JA+2).LE.0.) GOTO 4
      R1=YV(JA-1)
      Q1=YV(JA+2)
      T3=V(JA+1,JA-1)
      T4=V(JA+1,JA+2)
      CALL SOL44(XX,T,P1,Q1,P1,S1,T1,T2,T3,T4)
6    NT=2
      IX=IX+1
      X(TX)=XV(JA+1)
      Y(TX)=XX
      JA=JA+1
7    RETURN
      END

```

C SURROUTINE CD(C,D,T,XV,YV,X,Y,IA,JA,IX,NT,II,V)
C SEARCHES TOP SIDE OF RECTANGLE FOR GIVEN VALUE
C

```

      DIMENSION XV(1),YV(1),X(1),Y(1),V(100,200)
      IF (C.GT.T.AND.D.LT.T.OR.C.LT.T.AND.D.GT.T) 1,7
1    P1=XV(IA)
      Q1=XV(IA+1)
      XX=(P1+Q1)/2.
      T1=D
      T2=C
      IF (IA.EQ.1) 2,3
2    R1=XV(IA+2)
      T3=V(IA+2,JA+1)
      CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
      GO TO 6
3    IF (IA.EQ.II) 4,5
4    R1=XV(IA-1)
      T3=V(IA-1,JA+1)
      CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
      GO TO 6
5    IF (V(IA-1,JA+1).LE.0.) GOTO 2
      IF (V(IA+2,JA+1).LE.0.) GOTO 4
      R1=XV(IA-1)
      Q1=XV(IA+2)
      T3=V(IA-1,JA+1)
      T4=V(IA+2,JA+1)
      CALL SOL44(XX,T,P1,Q1,P1,S1,T1,T2,T3,T4)
6    NT=3

```

```

      TX=IX+1
      X(TX)=YX
      Y(TX)=YV(JA+1)
      JA=JA+1
7    RETURN
      END

C    SUBROUTINE DA(D,A,T,XV,YV,X,Y,IA,JA,IX,NT,JJ,V)
C    SEARCHES L.H. SIDE OF RECTANGLE FOR GIVEN VALUE
C
      DIMENSION XV(1),YV(1),X(1),Y(1),V(100,200)
      IF (D.GT.T.AND.A.LT.T.OR.D.LT.T.AND.A.GT.T) 1,7
1    P1=YV(JA)
      Q1=YV(JA+1)
      XX=(P1+Q1)/2.
      T1=A
      T2=D
      IF (JA.EQ.1) 2,3
2    R1=YV(JA+2)
      T3=V(IA,JA+2)
      CALL SOL33(XX,T,P1,Q1,P1,T1,T2,T3)
      GO TO 6
3    IF (JA.EQ.JJ) 4,5
4    R1=YV(JA-1)
      T3=V(IA,JA-1)
      CALL SOL33(XX,T,P1,Q1,R1,T1,T2,T3)
      GO TO 6
5    IF (V(IA,JA-1).LE.0.) GOTO 2
      IF (V(IA,JA+2).LE.0.) GOTO 4
      R1=YV(JA-1)
      S1=YV(JA+2)
      T3=V(IA,JA-1)
      T4=V(IA,JA+2)
      CALL SOL44(XX,T,P1,Q1,R1,S1,T1,T2,T3,T4)
6    NT=4
      TX=IX+1
      X(TX)=XV(IA)
      Y(TX)=YX
      IA=IA-1
7    RETURN
      END

C    SUBROUTINE SOL33(XX,T,P1,Q1,R1,T1,T2,T3)
C    POLYNOMIAL INTERPOLATION WITH 3 NODES
C
      XX=XX-P1
      Y1=0.
      Y2=Q1-P1
      Y3=R1-P1
      PQ=X2
      XX2=Y2*X2
      XX3=Y3*X3
      TT=T-T1
      TT2=T2-T1
      TT3=T3-T1
      DT=XX2*X3-YX3+Y2
      AA=(TT2*X3-TT3*X2)/DT
      AB=(XX2+TT3-YX3+TT2)/DT
      TX=TT
      TZ=TT2-TT

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

1  TY=(AA+XX+RR)*XX-TT
   TU=TX+TY
   IF (TU) 2,5,3
2  Y2=XX
   TZ=TY
   GO TO 4
3  Y1=XX
   TX=TY
4  XU=(Y1+X2)/2.
   U=ABS((XX-XU)/PQ)
   XX=XU
   IF (U,LT,.005) 5,1
5  XX=XX+P1
   RETURN
END

C
SUBROUTINE SOL44(XX,T,P1,Q1,R1,S1,T1,T2,T3,T4)
C
C POLYNOMIAL INTERPOLATION WITH 4 NODES
C
XX=XX-P1
Y1=0.
Y2=Q1-P1
Y3=R1-P1
Y4=S1-P1
PQ=X2
XX2=Y2+Y2
XX3=Y3+Y3
XX4=Y4+Y4
XXY2=XX2+X2
XXY3=XX3+X3
XXY4=XX4+X4
TT=T-T1
TT2=T2-T1
TT3=T3-T1
TT4=T4-T1
DT=DT3(XXX2,Y2,Y2,XXX3,Y3,X3,XXX4,Y4,X4)
AA=DT3(TT2,XX2,X2,TT3,XX3,X3,TT4,XX4,X4)/DT
RR=DT3(XXX2,TT2,Y2,XXX3,TT3,X3,XXX4,TT4,X4)/DT
CC=DT3(XXX2,Y2,TT2,XXX3,XX3,TT3,XXX4,XX4,TT4)/DT
TX=TT
TZ=TT2-TT
1  TY=((AA+XX+RR)*XX+CC)*XX-TT
   TU=TX+TY
   IF (TU) 2,5,3
2  Y2=XX
   TZ=TY
   GO TO 4
3  Y1=XX
   TX=TY
4  XU=(Y1+X2)/2.
   U=ABS((XX-XU)/PQ)
   XX=XU
   IF (U,LT,.005) 5,1
5  XX=XX+P1
   RETURN
END

C
FUNCTION DT3(A1,R1,C1,A2,R2,C2,A3,B3,C3)
C
C CALCULATES DETERMINANT OF 3 BY 3 MATRIX
C

```

DT3=A1*A2*C3+A2*A3*C1+A3*A1*C2-A1*A3*C2-A2*A1*C3-A3*A2*C1
RETURN
END

THIS PAGE IS BEST QUALITY PRACTICAL
FROM COPY FURNISHED TO DDC

```

PROGRAM BDYCHK

DIMENSION MX(100),MY(200),X(4),Y(4)
INTEGER BDR(199,10),BDZ(99,10)
DATA BLNK/4H /
101 FORMAT(10F8.0)
102 FORMAT(20I4)
110 FORMAT(1M1)
121 FORMAT(1H ,A1,3HMX(,13,3H) =,F8.3,4(A4,6X,3HMX(,13,3H) =,F8.3))
131 FORMAT(1H ,A1,3HMY(,13,3H) =,F8.3,4(A4,6X,3HMY(,13,3H) =,F8.3))
202 FORMAT(1H ,I4,5(4X,5I4))
READ 102,NX,NY,NEPR,NEPZ
READ 102,IO,IN,JU,JN
NXM1=NX-1
NYM1=NY-1
READ 101,(MX(I),I=1,NX)
READ 101,(MY(J),J=1,NY)
READ 102,((BDR(I,J),J=1,NEPR),I=1,NYM1)
READ 102,((BDZ(I,J),J=1,NEPZ),I=1,NXM1)
MX0=MX(IO)
DO 140 I=IO,IN
  MX(I+1-IO)=MX(I)
DO 140 IK=1,NEPZ
  IF (BDZ(I,IK).LT.2*JU) BDZ(I,IK)=2*JU
  IF (BDZ(I,IK).GT.2*JN) BDZ(I,IK)=2*JN
140 BDZ(I+1-IO,IK)=BDZ(I,IK)-2*(JU-1)
  NX=IN+1-IO
  NXM1=NX-1
  MY0=MY(JU)
DO 160 J=JU,JN
  MY(J+1-JU)=MY(J)-MY0
DO 160 JK=1,NEPR
  IF (BDR(J,JK).LT.2*IO) BDR(J,JK)=2*IO
  IF (BDR(J,JK).GT.2*IN) BDR(J,JK)=2*IN
160 BDR(J+1-JU,JK)=BDR(J,JK)-2*(IO-1)
  NY=JN+1-JU
  NYM1=NY-1
PRINT 110
K1=0
NCOL=NXM1/50
GOTO 222
220 NCOL=NCOL+1
222 INK=NCOL*50
225 K1=K1+1
K2=K1+INK
PRINT 121,(BLNK,K,MX(K),K=K1,K2,50)
IF (K1.EQ.50.OR.K1.EQ.NX) GOTO 229
IF (K2.EQ.NX) 220,225
229 PRINT 110
K1=0
NCOL=NYM1/50
GOTO 232
230 NCOL=NCOL+1

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
232 INK=NCUL*50
235 K1=K1+1
      K2=K1+INK
      PRINT 131,(BLNK,K,MY(K),K=K1,K2,50)
      IF (K1.EQ.50.OR.K1.EQ.NY) GOTO 239
      IF (K2.EQ.NY) 230,235
239 PRINT 110
      DO 250 J=1,NYM1
250 PRINT 202,J,(BDR(J,K),K=1,NEPR)
      PRINT 110
      DO 260 J=1,NXM1
260 PRINT 202,J,(BDZ(J,K),K=1,NEPZ)
      CALL PLOTS (11,0,99)
      CALL PLOT (0.,-10.,-3)
      CALL PLOT (0.,0.5,-3)
      XMAX=MX(NX)=MX(1)
      YMAX=MY(NY)=MY(1)
      FIRSTX=MX(1)
      DELTAX=XMAX/10.
      FIRSTY=MY(1)
      DELTAY=YMAX/10./XMAX
      CALL AXIS(0.,10.,6HZ=AXIS,+6,YAX,0.,FIRSTY,DELTAY)
      CALL AXIS(0.,10.,6HW=AXIS,-6,10.,270.,FIRSTX,DELTAX)
      X(3)=FIRSTX
      X(4)=DELTAX
      Y(3)=FIRSTY
      Y(4)=DELTAY
      DO 500 I=1,NYM1
      Y(1)=MY(I+1)
      Y(2)=MY(1)
      DO 400 J=1,NEPR
      IF (BDR(I,J).GT.2*NX) GOTO 500
      L=BDR(I,J)/2
      K=BDR(I,J)/2. + .5 - L
      X(1)=XMAX=MX(L)
      IF (K.GE.1) GOTO 350
      X(2)=X(1)
      GOTO 390
350 X(2)=XMAX=MX(L+1)
390 CALL LINE(Y,X,2,1,0,0)
400 IF (BDR(I,J).GE.2*NX) GOTO 500
500 CONTINUE
      DO 700 I=1,NXM1
      X(1)=XMAX=MX(I)
      X(2)=XMAX=MX(I+1)
      DO 600 J=1,NEPZ
      IF (BDZ(I,J).GT.2*NY) GOTO 700
      L=BDZ(I,J)/2
      K=BDZ(I,J)/2. + .5 - L
      Y(1)=MY(L)
      IF (K.GE.1) GOTO 550
      Y(2)=Y(1)
      GOTO 590
550 Y(2)=MY(L+1)
```

THIS PAGE IS BEST QUALITY REPRODUCTION
FROM COPY FURNISHED TO DDC

```

590 CALL LINE(Y,X,2,1,0,0)
600 IF (BDZ(I,J).GE.2*NY) GOTO 700
700 CONTINUE
    CALL PLOT(12,0,0,0,40)
    STOP
    END

```

PROGRAM EPSCHK

```

        DIMENSION R(99,199),Z(99,199),EPSR(199,10),EPSZ(99,10)
        INTEGER BDR(199,10),BDZ(99,10)
        DATA EPSR/1990*0.0/,EPSZ/990*0.0/
        DATA R/19701*-1.0/,Z/19701*-1.0/
102  FORMAT(20I4)
104  FORMAT(10F5,0)
150  FORMAT (1H1,3X,32HN      NX      NY  R(NX,NY)  Z(NX,NY))
155  FORMAT (1H ,315,2F10,3)
161  FORMAT (1H0,16X,2HNU)
162  FORMAT (1H0,12X,10HDIELECTRIC)
163  FORMAT (1H0,10X,15HINCONSISTENCIES)
175  FORMAT(1H1,5X,1H+,24(14,1X))
176  FORMAT(6X,1H+,24(14,1X))
180  FORMAT(6X,1H+,24(5H---+))
185  FORMAT(1X,14,2H +,24F5,1)
      READ 102,NX,NY,NEPR,NEPZ
      NXM1=NX-1
      NYM1=NY-1
      READ 102,((BDR(I,J),J=1,NEPR),I=1,NYM1)
2    READ 102,KU,KN,KK
      READ 104,(EPSR(KU,K),K=1,KK)
      IF (KU.EQ.KN) GOTO 4
      KI=KU+1
      DO 3 I=KI,KN
        DO 3 K=1,KK
3      EPSR(I,K)=EPSR(KU,K)
4      IF (KK.EQ.NEPR) GOTO 5
      KP=KK+1
      DO 45 I=KO,KN
        DO 45 J=KP,NEPR
45     EPSR(I,J)=EPSZ(I,KK)
5      IF (KN.LT.NYM1) GOTO 2
      READ 102,((BDZ(I,J),J=1,NEPZ),I=1,NXM1)
6      READ 102,KU,KN,KK
      READ 104,(EPSZ(KU,K),K=1,KK)
      IF (KU.EQ.KN) GOTO 8
      KI=KU+1

```

```

      DO 7 I=K1,KN
      DO 7 K=1,KK
      7 EPSZ(I,K)=EPSZ(KU,K)
      8 IF (KK.EQ.NEPZ) GOTO 9
      KP=KK+1
      DO 89 I=K0,KN
      DO 89 J=KP,NEPZ
      89 EPSZ(I,J)=EPSZ(I,KK)
      9 IF (KN.LT.NXM1) GOTO 6
      DO 20 I=1,NYM1
      M=0.
      K2=0
      DO 20 J=1,NEPM
      IF (M-K2.GT.0.) K2=K2+1
      K1=K2+1
      M=BDP(I,J)/2.-1.
      K2=M
      IF (K2.LT.K1) GOTO 20
      DO 10 K=K1,K2
      10 R(K,I)=EPSR(I,J)
      20 CONTINUE
      DO 40 I=1,NXM1
      M=0.
      K2=0
      DO 40 J=1,NEPZ
      IF (M-K2.GT.0.) K2=K2+1
      K1=K2+1
      M=BDZ(I,J)/2.-1.
      K2=M
      IF (K2.LT.K1) GOTO 40
      DO 30 K=K1,K2
      30 Z(I,K)=EPSZ(I,J)
      40 CONTINUE
      N=0
      PRINT 150
      DO 60 I=1,NXM1
      DO 60 J=1,NYM1
      IF (R(I,J).EQ.Z(I,J)) GOTO 60
      N=N+1
      PRINT 155,N,I,J,R(I,J),Z(I,J)
      IF (N.GE.1000) GOTO 70
      60 CONTINUE
      IF (N.GT.0) GOTO 70
      PRINT 161
      PRINT 162
      PRINT 163
      70 CONTINUE
      210 M2=0
      M1=50
      220 M2=M2+50
      IF (M2.LE.NY) GOTO 225
      M1=NY+50-M2
      M2=NY
      225 N2=0
      230 N1=N2+1

```

```

N2=N2+24
IF (N2.GT.NX) N2=NX
PRINT 175,(K,K=N1,N2)
PRINT 180
DO 250 J=1,M1
I=M2+1-J
250 PRINT 185,1,(R(K,I),K=N1,N2)
PRINT 180
PRINT 176,(K,K=N1,N2)
IF (N2.LT.NX) GOTO 230
IF (M2.LT.NY) GOTO 220
260 M2=0
M1=50
270 M2=M2+50
IF (M2.LE.NY) GOTO 275
M1=NY+50-M2
M2=NY
275 N2=0
280 N1=N2+1
N2=N2+24
IF (N2.GT.NX) N2=NX
PRINT 175,(K,K=N1,N2)
PRINT 180
DO 300 J=1,M1
I=M2+1-J
300 PRINT 185,1,(Z(K,I),K=N1,N2)
PRINT 180
PRINT 176,(K,K=N1,N2)
IF (N2.LT.NX) GOTO 280
IF (M2.LT.NY) GOTO 270
STOP
END

```

DISTRIBUTION LIST

ARMY DAEN-CWE-M (LT C D Binning), Washington DC; DAEN-FEU, Washington DC; ERADCOM Tech Supp Dir. (DELS-D) Ft. Monmouth, NJ; HQDA (DAEN-FEE-A); Tech. Ref. Div., Fort Huachuca, AZ

ARMY CORPS OF ENGINEERS Seattle Dist. Library, Seattle WA

NAVCOMMAREAMSTRSTA ENS F. Stearns, Wahiawa HI; PWO, Norfolk VA; SCE Unit I Naples Italy; SCE, Guam

NAVCOMMSTA CO (61E) Puerto Rico; CO, San Miguel, R.P.; PWO, Diego Garcia Island; PWO, Exmouth, Australia; PWO, Fort Amador Canal Zone; PWO, Puerto Rico

NAVFACENGCOM Code 0451 Alexandria, VA; Code 101 Alexandria, VA; Code 104 Alexandria, VA

NAVFACENGCOM - CHES DIV. Code 403 (H. DeVoe) Wash, DC; Code 405 Wash, DC; Code 406 (Meyer) Washington, DC

NAVFACENGCOM - LANT DIV.; Code 10A, Norfolk VA; Code 111, Norfolk, VA; NAS Norfolk, VA

NAVFACENGCOM - NORTH DIV. Code 111 (Castranovo) Philadelphia, PA

NAVFACENGCOM - PAC DIV. Commander, Pearl Harbor, HI

NAVFACENGCOM - WEST DIV. O9P/20

NAVFACENGCOM CONTRACT Dir. Eng. Div., Exmouth, Australia

NAVMIRO OIC, Philadelphia PA

NAVOCEANSYSCEN Code 6565 (Tech. Lib.), San Diego CA; Code 6700, San Diego, CA; Research Lib., San Diego CA; SCE (Code 6600), San Diego CA

NAVSCOLCECOFF C35 Port Hueneme, CA

NAVSEC Code 6034 (Library), Washington DC

NAVSECGRUACT PWO, Adak AK

NAVSHIPYD Code 202.4, Long Beach CA

NAVSTA CO Naval Station, Mayport FL

OFFICE SECRETARY OF DEFENSE OASD (MRA&L) Pentagon (T. Casberg), Washington, DC

ONR BROFF, CO Boston MA

NAVELEXSYSCOM Code 03, Washington DC

PMTIC Code 4253-3, Point Mugu, CA

PWC ACE Office (LTJG St. Germain) Norfolk VA; CO, Guam; Code 120, Oakland CA; Code 128, Guam; Code 200, Great Lakes IL; Code 40 (C. Kolton) Pensacola, FL; Code 42B (R. Pascua), Pearl Harbor HI

USCG (G-ECV/61) (Burkhart) Washington, DC

USCG ACADEMY LT N. Stramandi, New London CT

USCG R&D CENTER Tech. Dir. Groton, CT

USNA Ch. Mech. Engr. Dept Annapolis MD; PWD Engr. Div. (C. Bradford) Annapolis MD

CORNELL UNIVERSITY Ithaca NY (Serials Dept, Engr Lib.)

DAMES & MOORE LIBRARY LOS ANGELES, CA

MIT NROTC (Collins) Cambridge MA

NY CITY COMMUNITY COLLEGE BROOKLYN, NY (LIBRARY)

UNIV. NOTRE DAME Katona, Notre Dame, IN

SAN DIEGO STATE UNIV. Dr. Krishnamoorthy, San Diego CA

STANFORD UNIVERSITY Engr Lib, Stanford CA

UNIVERSITY OF CALIFORNIA Berkeley CA (E. Pearson)

UNIVERSITY OF DELAWARE Newark, DE (Dept of Civil Engineering, Chesson)

UNIVERSITY OF WISCONSIN Milwaukee WI (Ctr of Great Lakes Studies)

URS RESEARCH CO. LIBRARY SAN MATEO, CA

AUSTRALIA LNO, Army Mobile Equip R & D Com, Fort Belvoir VA

BECHTEL CORP. SAN FRANCISCO, CA (PHELPS)

CF BRAUN CO Du Bouchet, Murray Hill, NJ

COLUMBIA GULF TRANSMISSION CO. HOUSTON, TX (ENG. LIB.)

CONCRETE TECHNOLOGY CORP. TACOMA, WA (ANDERSON)

DURLACH, O'NEAL, JENKINS & ASSOC. Columbia SC

EVALUATION ASSOC. INC KING OF PRUSSIA, PA (FEDELE)

FORD, BACON & DAVIS, INC. New York (Library)

GOULD INC. Shady Side MD (Ches. Inst. Div., W. Paul)

GRUMMAN AEROSPACE CORP. Bethpage NY (Tech. Info. Ctr)

HUGHES AIRCRAFT Culver City CA (Tech. Doc. Ctr)

MCDONNELL AIRCRAFT CO. Dept 501 (R.H. Fayman), St Louis MO
RAND CORP. Santa Monica CA (A. Laupa)
SANDIA LABORATORIES Library Div., Livermore CA
SEATECH CORP. MIAMI, FL (PERONI)
TRW SYSTEMS REDONDO BEACH, CA (DAI)
WESTINGHOUSE ELECTRIC CORP. Annapolis MD (Oceanic Div Lib. Bryan)